

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zpracování obrazového signálu v obvodech s programovatelnou logikou
Image Signal Processing with Programmable Logic Devices

Zadání diplomové práce

Student:

Bc. Jan Kotyza

Studijní program:

N2649 Elektrotechnika

Studijní obor:

2601T004 Měřicí a řídicí technika

Téma:

Zpracování obrazového signálu v obvodech s programovatelnou logikou
Image Signal Processing with Programmable Logic Devices

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Vypracování přehledu standardů pro digitální přenos obrazové informace.
2. Nastudování standardu ITU-R BT.656.
3. Návrh koncepce demonstrační úlohy pro zpracování obrazového signálu s programovatelnou logikou FPGA.
4. Návrh logiky pro demonstrační úlohu v jazyce VHDL.
5. Implementace a oživení navržené logiky pro FPGA řady Spartan.
6. Experimentální ověření funkce a zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] PINKER, Jiří a Martin POUPA. *Číslicové systémy a jazyk VHDL*. 1. vyd. Praha: BEN - technická literatura, 2006. 349 s. ISBN 80-7300-198-5.
- [2] PARNELL, Karen and Nick MEHTA. *Programmable Logic Design Quick Start Handbook*. 4th ed. [s.l.]: Xilinx Inc., 2003. 225 s.
- [3] ASHENDEN, Peter J. *The Designer's Guide to VHDL*. San Francisco(USA): Morgan Kaufmann Publishers, 1999. 688 s. ISBN 1-55860-270-4.
- [4] KAŠÍK, Vladimír. *Programování hradlových polí*. Učební text a návody do cvičení. VŠB-TU Ostrava, 2012.
- [5] ŠŤASTNÝ, Jakub. *FPGA prakticky*. 1.vyd. Praha: BEN - technická literatura, 2010. 199s. ISBN 978-80-7300-261-9.
- [6] BOVIK, Alan C. *Handbook of Image and Video Processing*. 1.vyd. San Diego: Academic Press, c2000. 891 s. ISBN 0-12-119790-5.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

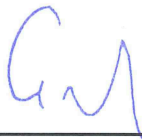
Vedoucí diplomové práce: **Ing. Vladimír Kašík, Ph.D.**

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 28. 4. 2016

podpis

Bc. Jan Mlýnský

Poděkování

Na tomto místě bych rád poděkoval Ing. Vladimíru Kašíkovi, Ph.D. za vedení diplomové práce, a také za jeho podporu, trpělivost, rady, inspiraci a diskuze nejen při vypracování této diplomové práce. Rovněž patří můj dík rodině za podporu při studiu a tvorbu potřebného zázemí.

Abstrakt

Práce obsahuje teorii pojednávající o přenosových standardech obrazové informace. Jsou zde uvedeny standardy popisující přenos informací mezi kamerou a nadřazeným systémem, standardy používané při přenosu informací mezi jednotlivými integrovanými obvody a standardy používané pro kódování přenášených dat. Dále je zde obsažen stručný přehled operací využívaných při zpracování obrazu. Po návrhu konceptu úlohy je zde popsán realizovaný systém s programovatelnou logikou, který implementuje rozhraní dekodující vstupní data a jednoduché operace pro zpracování obrazu. Součástí práce jsou také dokumenty umístěné v přílohách, popisující několik vybraných VHDL modulů.

Abstract

The thesis contains a theory explaining transmission standards of image information. Standards listed here describe transfer of data between the camera and the master system, the standards used in data transmission between integrated circuit and the standards used for coding of transmitted data. Included is also a short overview of the operations used in the image processing. After designing the concept of task there is a description of implemented system with a field programmable gate array, which implements the interface decoding input data and simple operation for image processing. The work also contains documents located in the annexes, describing several selected VHDL modules.

Klíčová slova

Programovatelná hradlová pole, zpracování obrazu, ITU-R BT.656, FFPGA, odstíny šedé, odstíny hnědé, černobílý obraz, prahování, hranový detektor, standard.

Key Words

Field programmable gate arrays, image processing, ITU-R BT.656, FPGA, greyscale, sepia, binary image, threshold, edge detect, standard.

Obsah

Zkratky	9
Cizí slova.....	10
Jednotky a použité znaky	10
Seznam ilustrací	11
Seznam tabulek	12
Úvod.....	13
1 Digitální přenos obrazové informace	14
1.1 Přenos mezi zařízeními	14
1.1.1 USB 3.0 Vision	14
1.1.2 Firewire (IEEE 1394).....	14
1.1.3 HDMI	15
1.1.4 Gigabite Ethernet Vision.....	15
1.1.5 Camera Link.....	16
1.1.6 CoaXPress	17
1.2 Přenos obrazové informace mezi integrovanými obvody	17
1.3 Formáty digitálního datového přenosu.....	18
1.3.1 ITU-R BT.656-5	18
1.3.2 ITU-R BT.799-4.....	18
1.3.3 ITU-R BT.1120-8.....	18
1.3.4 ITU-R BT.1303	19
1.3.5 MIPI	19
1.3.6 Podvzorkování.....	19
1.4 Barevné modely.....	20
1.4.1 Model RGB	20
1.4.2 Model YUV.....	21
1.4.3 Model CbYCr.....	21
2 Digitální zpracování obrazu	22
2.1 Úprava obrazu	22
2.1.1 Převod do odstínů šedé barvy.....	22
2.1.2 Histogram.....	22
2.1.3 Úprava jasu.....	23
2.1.4 Inverze obrazu	23
2.2 Segmentace obrazu.....	24
2.2.1 Prahování.....	24
2.2.2 Detekce hran.....	24

2.3	Morfologické operace	25
2.3.1	Dilatace.....	25
2.3.2	Eroze	25
3	Použitý standard ITU-R BT.656	26
3.1	Formát dat standardu ITU-R BT.656	26
4	Koncepce úlohy.....	28
4.1	Kamera Canon MV530	29
4.1.1	CCIR PAL signál	29
4.1.2	Kompozitní video.....	29
4.2	ADV modul	30
4.3	Vývojový kit Nexys 3	33
4.4	TFT displej	34
4.4.1	Dotyková vrstva	36
5	Návrh logiky.....	39
5.1	ITU-BT.656 dekodér.....	39
5.2	Detekce dat formátu CbYCr.....	42
5.3	Zpracování obrazu.....	42
5.3.1	Převod do RGB	43
5.3.2	Převod do odstínů šedé.....	44
5.3.3	Převod do odstínů hnědé (sépie)	45
5.3.4	Převod do binárního obrazu	46
5.3.5	Hranový detektor.....	47
5.4	TFT řadič.....	47
5.4.1	Napájení TFT displeje.....	47
5.4.2	Časová základna.....	48
5.5	Vyrovňovací paměť.....	49
5.6	Uživatelské rozhraní pro ovládání úlohy.....	50
5.6.1	Realizace tlačítek na dotykové vrstvě displeje.....	51
6	Diagnostika vybraných modulů v FPGA	52
6.1	Simulace.....	52
6.2	Testování	53
6.3	Syntéza	54
7	Zhodnocení využití FPGA při zpracování obrazu.....	55
	Závěr	56
	literatura	57
	Přílohy	59

Zkratky

zkratka	anglický překlad	český překlad
BT	broadcasting service	rozhlasové vysílání
CCD	charge coupled device	zařízení s vázanými náboji
CCIR	consultative Committee on International Radio	poradní komise pro mezinárodní rozhlas
CML	current mode logic	režim proudové logiky
CMOS	complementary metal oxide semiconductor	polovodič doplněný oxidem kovu
CMYK	cyan, magenta, yellow, key (black)	azurová, purpurová, žlutá, černá
DAC	digital to analog converter	převodník z digitálního do analogového signálu
DCM	digital clock manager	manažer digitálních hodin
EAV	the end of active video signal	konec aktivního videa
ECL	emitter coupled logic	emitory vázaná logika
FIFO	first in, first out	první dovnitř, první ven
FPGA	field programmable gate array	programovatelná hradlová pole
HD	high definition	vysoké rozlišení
IO	integrated circuit	integrovaný obvod
IP	Intellectual property	duševní vlastnictví
ISO / OSI	international Standards Organization / Open system Interconnection	mezinárodní normalizační organizace / propojení otevřeného systému
LFCSP	lead frame chip scale packed	označení typu pouzdra obvodu
LSB	least significant Bit	nejméně významný Bit
LUT	look up table	nahlížecká tabulka
LVDS	low voltage differential signaling	nízkonapěťový rozdílový signál
LVPECL	low voltage positive emitter coupled logic	kladná nízkonapěťová emitory vázaná logika
LVTTL	low voltage transistor transistor logic	nízkonapěťová tranzistorově tranzistorová logika
MSB	most significant Bit	nejvýznamnější Bit
NTSC	national television system committee	národní komise pro televizní systém
PAL	phase alternation line	s fází se střídající řádek
PC	personal computer	osobní počítač
PCB (DPS)	printed circuit board	deska plošných spojů
PECL	positive emitter coupled logic	kladná emitory vázaná logika
PoE	power over Ethernet	napájení přes Ethernet
PWM	pulse width modulation	pulzně šířková modulace
RAM	random access memory	paměť s přímým přístupem
SAV	the start of active video signal	začátek aktivního videa
TFT	thin film transistor	tenká vrstva s tranzistory
TTL	transistor transistor logic	tranzistorově tranzistorová logika
UART	universal asynchronous receiver and transmitter	univerzální asynchronní přijímač a vysílač
USB	universal serial bus	univerzální sériová sběrnice
VGA	video graphics array	označení grafického rozhraní

VHDCI	very high density cable interconnect	kabel s vysokou hustotou propojení
VHDL	VHSIC hardware description language	jazyk pro popis VHSIC
VHSIC	very high speed integrated circuit	velmi rychlé integrované obvody
VML	voltage mode logic	režim napěťové logiky

Cizí slova

Cizí slovo	Český překlad
additive	aditivní - sčítání
anti-aliasing	vyhlazování hran
base	základní
buffer	vyrovnávací paměť
composite	kompozitní
core	jádro
enabled	povolení
false contouring	falešné konturování
field	pole
frame	rámec
full	plná
latency	zpoždění
line	čára
medium	střední
preprocessing	před zpracování
rate	frekvence
real time	reálný čas
subtractive	subtraktivní - rozdílové
time	čas

Jednotky a použité znaky

Jednotka	Základní jednotka	význam
Mb/s	b/s	bit za sekundu
MHz	Hz	kmitočet
V	V	elektrické napětí
bpp	bpp	bity na bod
fps	fps	počet snímků za sekundu
m	m	délka

znak	význam
*	součin
x	absolutní hodnota z x
\oplus	exkluzivní disjunkce
°	stupně

Seznam ilustrací

Obr. 1 USB 3.0 Vision konektor a kamera [2].....	14
Obr. 2 Firewire konektor [4] a kamera [6]	15
Obr. 3 HDMI rozhraní [5] a kamera [9].....	15
Obr. 4 Gigabite Ethernet rozhraní [2]a kamera [7]	16
Obr. 5 Camera Link rozhraní [3] kamera [8]	16
Obr. 6 CoaXPress rozhraní [10] a kamera [11].....	17
Obr. 7 Podvzorkování obrazu.....	19
Obr. 8 Barevné modely, vlevo model RGB, vpravo model CMYK [21].....	20
Obr. 9 Snímek v odstínech šedé barvy	22
Obr. 10 Histogram vzorového snímku	23
Obr. 11 Úprava jasu vzorového snímku vlevo, histogram světlého snímku vpravo	23
Obr. 12 Invertovaný vzorový snímek vlevo, histogram vpravo	23
Obr. 13 Binární snímek	24
Obr. 14 Detekce hran	24
Obr. 15 Snímek s aplikovanou dilatací	25
Obr. 16 Snímek s aplikovanou erozí	25
Obr. 17 Paralelní přenos dat ve formátu BT.656 [1].....	26
Obr. 18 Rozložení jednoho snímku [1]	27
Obr. 19 Blokové schéma znázorňující koncept úlohy.....	28
Obr. 20 Kamera MV550 [17].....	29
Obr. 21 Blokové schéma IO ADV 7180 [18].....	30
Obr. 22 DPS modulu s ADV 7180 vlevo TOP vpravo BOTTOM vrstva [20]	31
Obr. 23 DPS pro připojení modulu s ADV7180 ke kitu Nexys 3	31
Obr. 24 Vývojový kit Nexys 3 [16].....	33
Obr. 25 Blokové schéma vývojového kitu Nexys 3 [16]	33
Obr. 26 Vmod TFT displej [15]	34
Obr. 27 Časování TFT displej [15]	35
Obr. 28 Význam signálu pro zobrazení na displeji	36
Obr. 29 Schéma zapojení měřícího obvodu dotykové vrstvy	37
Obr. 30 Sériová komunikace s ADV7873 [23].....	37
Obr. 31 Stavový registr obvodu ADV7873 [23].....	37
Obr. 32 Stavový diagram znázornění funkce detekce dat	40
Obr. 33 Záznam vstupních dat z analyzátoru ChipScope pro znázornění funkce modulu BT656.....	41
Obr. 34 Diagram stavového automatu pro detekci složek Cb Y Cr	42
Obr. 35 Separace jednotlivých složek	42
Obr. 36 Symbol modulu VHDL pro převod CbYCr do RGB	44
Obr. 37 Symbol modulu VHDL pro převod RGB do odstínů šedé.....	45
Obr. 38 Symbol modulu VHDL pro převod RGB do odstínů hnědé	46
Obr. 39 Symbol modulu VHDL pro převod do binárního obrazu	46
Obr. 40 Symbol modulu VHDL pro hranový detektor	47
Obr. 41 Blokové uspořádání PWM generátoru.....	48
Obr. 42 Blokové schéma stavového automatu pro řízení napájení LCD displeje.....	48
Obr. 43 Symbol modulu VHDL vyrovnávací paměť	49
Obr. 44 Ovládací menu	50
Obr. 45 Čtyřmístný sedmisegmentový displej	51
Obr. 46 Informační rozložení displeje.....	51
Obr. 47 Prostředí ISim	52
Obr. 48 Prostředí ChipScope.....	54

Seznam tabulek

Tab. 1 Prostředky pro návrh.....	13
Tab. 2 Konfigurace camera Link.....	16
Tab. 3 Napěťové úrovně TTL a CMOS (14).....	17
Tab. 4 Napěťové úrovně rozdílových standardů (12), (13).....	17
Tab. 5 Devíti stavová logika	18
Tab. 6 Popis dat při SAV (XY) a EAV (XY) sekvenci (1).....	27
Tab. 7 Rozložení jednoho snímku [1]	27
Tab. 8 Parametry video signálu kamery MV550	29
Tab. 9 Propojení ADV7180 s Nexys 3.....	32
Tab. 10 Typické parametry TFT displeje [15]	34
Tab. 11 Odpor dotykové vrstvy.....	36
Tab. 12 12 bitové hodnoty pro kraje dotykové vrstvy [15].....	36
Tab. 13 Popis stavového registru obvodu ADV7873 [23]	38
Tab. 14 SAV a EAV byty.....	39
Tab. 15 Koeficienty pro přepočít sloček CbYCr na RGB	43
Tab. 16 Koeficienty pro přepočít sloček R G B do odstínů šedé.....	44
Tab. 17 Koeficienty pro přepočít sloček R G B do odstínů hnědé	45
Tab. 18 Registr pro detekci hrany	47
Tab. 19 Verze ISim	53
Tab. 20 Verze ChipScope.....	53

Úvod

Při vyhledávání objektů v obraze je často vstupní obraz složitě upravován a převáděn do takové podoby, aby vyhledávací algoritmy byly co nejefektivnější. Tyto úkony spadají do tzv. předzpracování obrazu (preprocessing). V předzpracování se běžně provádí rozostřování obrazu, korekce jasu, úprava rozlišení, aplikují se různé filtry, obraz se převádí do odstínů šedi, poté se převádí na binární obraz a další operace. Některé operace snižují obsah nepotřebných dat a zrychlují celý proces, jiné zvyšují kvalitu zájmových oblastí pro další zpracování. Většina úloh, ve kterých se obraz zpracovává, jsou limitované časem a je tudíž snaha tyto úlohy co nejvíce zrychlit. Využití programovatelné logiky nabízí možnost ulehčit běžnému PC práci a převzít některé části procesu.

Návrh realizovaný v této práci demonstruje příklady zpracování obrazu na obvodech s programovatelnou logikou. Obvody FPGA nabízejí poměrně velkou možnost paralelního zpracování dat. Z toho důvodu není nutné vyžadovat po návrhu vysoké pracovní frekvence, jak je tomu u standardních počítačů, ale řádově nižší. I tak mohou být obvody FPGA s řádově nižší frekvencí při zpracování řádově výkonnější, než běžně použitý počítač.

Zdroj signálu použitý v úloze je realizovaný komerčně dostupnou videokamerou. Zpracovaný signál je přiváděn a následně vykreslován na displej. Při praktickém použití je možné změnit zdroj vstupních informací, např. používat obrazová data přímo z výkonného CCD snímače, a data např. posílat k dalšímu zpracování do počítače přes libovolné rozhraní, kde by byl spuštěn algoritmus např. pro detekci objektů v obraze.

Po realizaci návrhu pro FPGA je zvoleno několik vybraných bloků, které by mohly být v budoucnu využitelné například pro jiné projekty a byly by snadno začleněny do rozsáhlejšího algoritmu. Ke každému vybranému bloku byl vytvořen dokument, ve kterém jsou sepsány všechny potřebné informace pro možnou implementaci bloku do složitějšího projektu, jeho testování a simulaci.

Veškeré práce jsou realizovány ve vývojovém prostředí dodávaného výrobcem FPGA Xilinx. FPGA použité k realizaci je od stejnojmenného výrobce viz tab. 1.

Tab. 1 Prostředky pro návrh

	jméno	verze	vydání / revize	datum aktualizace
software	ISE Design Suite	13.1 (nt64)	0.40d	28. únor, 2011
hardware	Nexys	3	B	10. duben, 2013

	jméno	verze	označení	pouzdro
FPGA	Spartan	6	XC6SLX16	CSG324C

1 Digitální přenos obrazové informace

Přenos digitální obrazové informace je prováděn na několika úrovních. První a asi nejznámější je přenos mezi zařízeními, kdy jsou k dispozici kamera a zařízení, jenž jsou propojeny (např. s počítačem, vestavným systémem, počítačovou sítí). S dalšími standardy se lze setkat, bude-li nutné přenášet informace mezi digitálními obvody v samotném zařízení. Tvar, jaký mají přenášená data mezi integrovanými obvody na zpracování obrazu (např. z CCD čipů) a ostatními částmi systému, je definován dalšími standardy. Tyto standardy jsou formáty datového přenosu.

1.1 Přenos mezi zařízeními

V této kapitole je provedeno srovnání standardů pro digitální přenos mezi kamerovým systémem a zpracovávajícím zařízením. Existuje mnoho způsobů. Některé z nich jsou považovány za standardní, jiné se používají ve specifických případech. Níže je nastíněn stručný popis několika nejznámějších.

1.1.1 USB 3.0 Vision

Standard USB3 Vision

Představen v lednu roku 2013. Stal se standardem pro strojové vidění. Je možné s jeho pomocí připojit kameru až 100 m dlouhým vodičem. Nabízí nízké hodnoty latence při ovládání kamer a přenosu dat. Dobře využitelný v real time aplikacích. Lze dosáhnout přenosové rychlosti až 350 Mb/s, velmi rozšířené rozhraní u počítačů obr. 1. Lze napájet zařízení z USB. Je obtížné připojit více kamer do jednoho systému.



Obr. 1 USB 3.0 Vision konektor a kamera [2]

1.1.2 Firewire (IEEE 1394)

Standard DCAM

Známý také jako IEEE 1394 obr. 2. Dnešní verze Firewire jsou standardizovány, díky čemuž poskytují možnost využívat toto rozhraní v pestré škále aplikací. Standard poskytuje velmi stabilní metodu pro výměnu dat mezi kamerami a počítačem. Z toho důvodu lze předvídat dobu přenosu a nedochází k chybám. Nízká latence a zátěž CPU. Lze dosáhnout rychlosti přenosu 32MB/s u IEEE 1394a nebo 64MB/s u IEEE 1394b.



Obr. 2 Firewire konektor [4] a kamera [6]

1.1.3 HDMI

Rozhraní často využívané u většiny moderní zobrazovací elektroniky obr. 3. Lze s jeho pomocí přenášet obraz i zvuk. HDMI je schopné přenášet obraz v rozlišení až 4K / ultra HD až 60 FPS. Problém nastává při přenosu na velké vzdálenosti. Běžné kabely jsou dostupné do délky 10 m. Při zakázkové výrobě lze dosáhnout až 23 m při použití opakovacího 46 m.



Obr. 3 HDMI rozhraní [5] a kamera [9]

1.1.4 Gigabite Ethernet Vision

Standard GigE Vision

Nejrychleji se rozvíjející rozhraní v oblasti průmyslového zpracování obrazu. Lze dosáhnout přenosové rychlosti 120 Mb/s při délce vedení 100 m. Připojené kamery mohou využívat Power over Ethernet (PoE) obr. 4. Lze využívat již existující ethernetové infrastruktury a připojit na síť více kamer. Využívá standardu GigE Vision a GenIcam, z toho důvodu je možné kombinovat kamery od různých výrobců v jedné síti.



Obr. 4 Gigabite Ethernet rozhraní [2] a kamera [7]

1.1.5 Camera Link

Standard Camera Link

Extrémně robustní a výkonné rozhraní pro průmyslové kamery. S rozhraním Camera Link obr. 5 lze připojit i velmi malé kamery. Rozhraní bylo speciálně navrženo pro průmyslové kamery. Lze dosáhnout přenosové rychlosti od 100 Mb/s až do cca 850 Mb/s tab. 2. Existují tři rychlostní konfigurace base, medium, full.

Tab. 2 Konfigurace camera Link

	rychlost [MB/s]	data	připojení
base	255	24 bitů	jeden kabel
medium	510	48 bitů	dva kabely
full	850	85 bitů	dva kabely



Obr. 5 Camera Link rozhraní [3] kamera [8]

1.1.6 CoaXPress

Standard nové generace. Využívá standardního koaxiálního kabelu. Ty mohou dosahovat více než 100 m. Kamery připojené přes CoaXPress obr. 6 lze synchronizovat s velmi nízkou latencí. Komunikační rychlost dosahuje až na 780 MB/s, při agregaci linky lze ale dosáhnout až 3 GB/s.



Obr. 6 CoaXPress rozhraní [10] a kamera [11]

1.2 Přenos obrazové informace mezi integrovanými obvody

Pro přenos dat mezi jednotlivými integrovanými obvody jsou k dispozici další spousty standardů. Popisují napěťové úrovně. Při návrhu složitějších zařízení je vhodné tyto standardy znát a brát v úvahu jejich vzájemnou kompatibilitu. V základě je lze rozdělit do dvou skupin. V první jsou standardy, které přenášejí informaci (log 1 a log 0) pomocí napětí vztahenému k nulovému potenciálu. Jsou popsány v tab. 3. Druhou skupinou používanou v zařízeních, kde záleží na spotřebě, na okolním rušení, kde se přenášejí velké objemy dat, jsou standardy v tab. 4. Tyto standardy jsou založeny na rozdílu nízkonapěťových signálů.

Tab. 3 Napěťové úrovně TTL a CMOS (14)

		5V TTL	5V CMOS	3,3V LVTTTL	2,5V CMOS	1,8V CMOS
vstup	H _{max}	5,00	5,00	3,30	2,50	1,80
	H _{min}	2,00	3,50	2,00	1,70	1,20
	L _{max}	0,80	1,50	0,80	0,70	0,65
	L _{min}	0,00	0,00	0,00	0,00	0,00
výstup	H _{max}	5,00	5,00	3,30	2,50	1,80
	H _{min}	2,40	4,44	2,40	2,00	1,45
	L _{max}	0,40	0,50	0,40	0,40	0,45
	L _{min}	0,00	0,00	0,00	0,00	0,00

Tab. 4 Napěťové úrovně rozdílových standardů (12), (13)

		LVDS	3,3V LVPECL	PECL	ECL	VML	CML
výstup	H _{min}	1,45	2,35	4,10	4,10	1,65	1,90
	L _{max}	1,07	1,60	3,20	3,20	0,85	1,10

Tyto standardy popisují logické úrovně, vyskytují se v běžné elektronice, nicméně při používání programovatelné logiky se můžeme setkat s dalšími stavy viz tab. 5. Jsou využívány převážně pro účely simulace. Význam jednotlivých úrovní vyplývá z níže uvedené tabulky.

Tab. 5 Devíti stavová logika

úroveň	stav	význam
U	nedefinovaný stav	často, není-li nastavená počáteční hodnota při simulaci
X	tvrdá neznámá hodnota	na pin jsou přivedeny dvě rozdílné tvrdé logické úrovně
0	tvrdá logická 0	přízemněný pin
1	tvrdá logická 1	napětí připojené na pin ze zdroje
Z	stav vysoké impedance	odpojený pin
W	slabá neznámá hodnota	na pin jsou přivedeny dvě rozdílné slabé logické úrovně
L	slabá logická 0	logická úroveň připojena přes rezistor k zemi (Pull-down)
H	slabá logická 1	logická úroveň připojena přes rezistor k napětí (Pull-up)
-	není důležitý	není důležité znát hodnotu

1.3 Formáty digitálního datového přenosu

Formátů pro přenos obrazové informace mezi obvody je k dispozici velké množství. Liší se především v rozlišení, rychlosti, popřípadě způsobem podvzorkování. Lze je najít jak u jednoduchých převodníků, tak i u profesionálních televizních videokamer. Níže je uvedena pouze část dostupných formátů.

1.3.1 ITU-R BT.656-5

Formát zavedený v roce 2007. Popisuje strukturu paralelních a sériových přenášených dat pro 525 a 625 řádkový digitální signál. Velikost makro pixelu je 4:2:2. Data jsou zakódována v 8 a 10 bitových slovech.

- video signál
- digitální zatmívací signál
- referenční časový signál
- pomocné datové signály

1.3.2 ITU-R BT.799-4

Formát zavedený v roce 2007. Popisuje strukturu sériově přenášených dat pro 525 a 625 řádkový digitální signál. Velikost makro pixelu je 4:4:4. Data jsou zakódována v 10 bitových slovech.

- video signál
- digitální zatmívací signál
- referenční časový signál
- pomocné datové signály

1.3.3 ITU-R BT.1120-8

Formát zavedený v roce 2012. Popisuje strukturu sériového přenosu. Lze přenášet HD obraz. Rozsahy řádku 1 125 a 1 080 se snímkovací frekvencí 24, 25, 30, 50 a 60 Hz. Velikost makro pixelu je 4:2:2, 4:4:4, 4:2:2:4, 4:4:4:4.

- video signál (10 bitové slova)
- referenční časový signál a identifikace kódu (10 bitové slova)

- pomocné datové signály

1.3.4 ITU-R BT.1303

Formát zavedený v roce 1997. Rozhraní popisuje strukturu paralelních a sériových přenášených dat pro 525 a 625 řádkový digitální signál. Velikost makro pixelu je 4:2:2 nebo 4:4:4. Data jsou zakódována v 8 a 10 bitových slovech.

- video signál
- digitální zatmívací signál
- referenční časový signál
- pomocné datové signály

1.3.5 MIPI

Jedná se o průmyslový rozhraní popisující přenos dat. Je zaměřený na mobilní zařízení jako jsou chytré telefony, tablety nebo netbooky. Standart pokrývá přenos dat po sériových sběrnicích. Dále definuje způsoby pro synchronizaci, rozhraní pro řízení kamer a v neposlední řadě také datové formáty. [24]

1.3.6 Podvzorkování

Podvzorkování obr. 7 se provádí z důvodu komprese obrazových dat. Využívá toho, že lidské oko je na odstíny méně citlivé, než na intenzitu světla. Z toho důvodu při přenosu obrazu ve formátu CbYCr lze dosáhnout snížením bitové hloubky Cb a Cr a značně snížit náročnost na systém. Podvzorkováním se vytvoří tzv. makro pixel se šířkou čtyř pixelů a výškou dvou řádků. Podvzorkování vznikne, jsou-li nahrazeny čtyři pixely v horním nebo dolním řádku společnou hodnotou barevné složky. Udává se ve formátu $X:Y:Z$, kde X je šířka makro pixelu, která je vždy 4. Y je počet barvonosných signálů v horním řádku (4 – každý pixel má svoji barvonosnou složku, 1 – čtyři barvonosné složky jsou nahrazeny jednou průměrnou hodnotou) a Z udává počet signálů na dolním řádku.

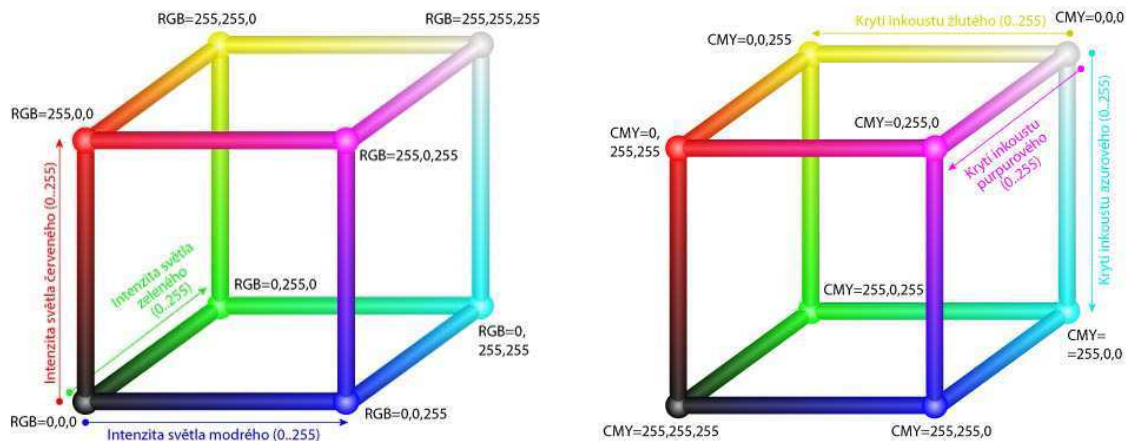


Obr. 7 Podvzorkování obrazu

- a) nepodvzorkovaný makro pixel (4:4:4)
- b) poloviční podvzorkování (4:2:2)
- c) čtvrtinové podvzorkování (4:1:1)

1.4 Barevné modely

V barevném světě jsou dvě možnosti míchání barev, a to aditivně, barevné složky se sčítají a výsledná barva má větší intenzitu. Tento princip je obdobný jako skládání světla. Využívá se v monitorech nebo projektorech (RGB). Druhou možností je subtraktivní míchání barev, která na rozdíl od předchozího způsobu světlo ubírá. Každá barva světlo více nebo méně pohlcuje. Toho je využito například u tiskáren (CMYK). Na obrázku obr. 8 jsou znázorněny oba barevné modely. V každém rohu krychle je jedna barva. Nastavení hodnot na obrázku RGB modelu v levém dolním rohu naznačuje princip skládání barev. V případě, kdy budou nastaveny všechny na 255, po uhlopříčce krychle se lze dostat na barvu bílou. To platí i pro model CMYK. Modelů je více, ale zde jsou popsány pouze ty nejznámější.



Obr. 8 Barevné modely, vlevo model RGB, vpravo model CMYK [21]

1.4.1 Model RGB

Nejznámější barevný model. Využívá zobrazení tří barevných složek, na které je oko nejvíce citlivé - červené, zelené a modré. Existuje také varianta RGBA, kde A znamená alfa, která reprezentuje průhlednost. Využívána především v počítačové grafice. Při převodu z ostatních barevných modelů lze postupovat podle rovnice (1.1) pro převod z barevného modelu CbYCr nebo (1.2) pro převod z modelu YUV. Rovnice jsou převzaty z [19].

Složky R' G' B' jsou barevné složky, které prošly tzv. Gama korekcí. To znamená, že rozsah složek je od 0 do 1. Index 601 značí jasovou složku použitou u standardu ITU-BT 601, který využívá množství standardů ITU-BT xxx.

$$\begin{aligned} R' &= Y_{601} + 1,371 * (Cr - 128) \\ G' &= Y_{601} - 0,689 * (Cr - 128) - 0,336 * (Cb - 128) \\ B' &= Y_{601} + 1,732 * (Cr - 128) \end{aligned} \quad (1.1)$$

$$\begin{aligned} R' &= Y + 1,140 * V \\ G' &= Y - 0,395 * U - 0,581 * V \\ B' &= Y + 2,032 * U \end{aligned} \quad (1.2)$$

1.4.2 Model YUV

Barevný model využívaný v televizní technice. Je rozdělen na jasovou a barevnou složku. Jasová složka Y nese informaci o černé a bílé. Barvonosné složky U a V nesou dopočet pro zbylé barvy, aby výsledný snímek byl barevný. Jednotlivé složky lze dopočítat dle rovnice (1.3) převzaté z [19].

$$\begin{aligned}Y &= 0,299 * R' + 0,587 * G' + 0,114 * B' \\U &= -0,147 * R' + 0,289 * G' + 0,436 * B' = 0,492 * (B' - Y') \\V &= 0,615 * R' + 0,515 * G' + 0,100 * B' = 0,877 * (R' - Y')\end{aligned}\tag{1.3}$$

1.4.3 Model CbYCr

Barevný model využívaný jako video standard. Jedná se o posunutou verzi formátu YUV. CbYCr je vhodnější pro přenos číslicovou technikou. Rozsah jednotlivých složek je definován jako 8 bitová hodnota, a to v rozsahu od 16 do 235. Rozsah je snížen z důvodu odstranění šumu. Jednotlivé složky lze vypočítat dle rovnice (1.4), která je převzata z [19].

$$\begin{aligned}Y_{601} &= 0,299 * R' + 0,587 * G' + 0,114 * B' \\Cb &= -0,172 * R' - 0,339 * G' + 0,511 * B' + 128 \\Cr &= 0,511 * R' + 0,428 * G' + 0,083 * B' + 128\end{aligned}\tag{1.4}$$

2 Digitální zpracování obrazu

Zpracováním obrazu se rozumí libovolná signálová operace, jejíž vstup je snímek a výstup opět snímek nebo množina dat. Při zpracovávání obrazu se pracuje se snímkem jako dvourozměrnou maticí dat, v případě barevného snímku jako tří dvourozměrných matic. Vstupní data mohou pocházet z libovolného zdroje, například kamery, fotoaparátu, ultrazvuku atd. Cíl zpracování závisí na konkrétní aplikaci. Níže jsou uvedeny pouze vybrané operace z nepřeberného množství.

2.1 Úprava obrazu

Na počátku každého zpracování je vhodné, aby vstupní data byla dostatečně kvalitní. Tím se rozumí, aby rozlišení, ostrost nebo jas odpovídaly zamýšlenému zpracování.

2.1.1 Převod do odstínů šedé barvy

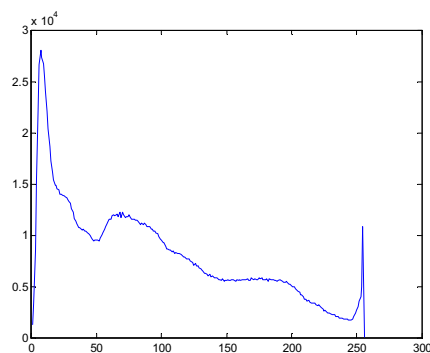
Převod do odstínů šedé barvy se provádí často z důvodu snížení datového toku nebo k zjednodušení následujících operací, popřípadě je-li snímek v odstínech šedé barvy vhodnější pro další zpracování. Při převodu dochází k trvalé ztrátě dat. Převod probíhá na základě rovnice (5.5), respektive tab. 16.



Obr. 9 Snímek v odstínech šedé barvy

2.1.2 Histogram

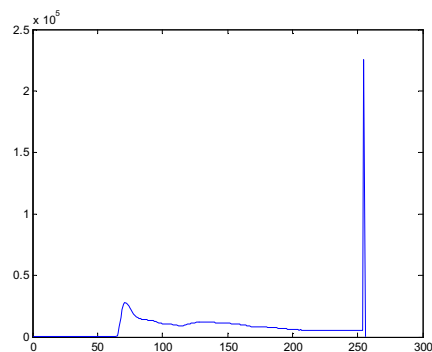
Úpravy obrazu se provádí na základě tzv. histogramu. Jedná se o rozložení jednotlivých pixelů v rozsahu barevné hloubky snímku. Histogram je převážně získáván ze snímků v odstínech šedé barvy. Je-li originální snímek v odstínech šedé barvy s 8 bitovým rozlišením, bude histogram tvořit 256 hodnot (odstínů). Každá hodnota bude reprezentovat množství pixelů zastoupených daným odstínem. Na obr. 10 je znázorněný histogram z obr. 9.



Obr. 10 Histogram vzorového snímku

2.1.3 Úprava jasu

Při nevhodných podmínkách při pořizování snímku je vhodné využití úpravy jasu snímku. Realizace probíhá přičtením konstanty k matici dat obr. 11. V závislosti jestli se bude jednat o kladnou nebo zápornou konstantu, bude výsledný snímek světlejší nebo tmavší.

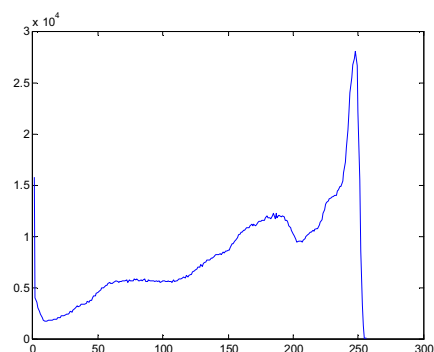


Obr. 11 Úprava jasu vzorového snímku vlevo, histogram světlého snímku vpravo

2.1.4 Inverze obrazu

Další možností úpravy snímku je inverze obr. 12. Výpočet je proveden dle rovnice (2.1). Kde $snimek_{inv}$ je výsledný invertovaný snímek a $snimek$ je originální snímek. Využití je opět závislé dle aplikace.

$$snimek_{inv} = 255 - snimek \quad (2.1)$$



Obr. 12 Invertovaný vzorový snímek vlevo, histogram vpravo

2.2 Segmentace obrazu

Při segmentaci obrazu dochází k členění souvisejících částí objektů na snímku. Výsledkem je snímek se shodným rozlišením jako vstupní snímek. Mezi objekty je vyznačená hranice, která objekty odděluje.

2.2.1 Prahování

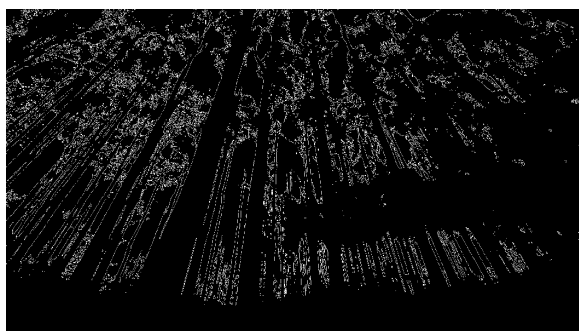
Operace sloužící k rozdělení pixelů ve snímku podle prahovací úrovně. Výsledkem je černobílý snímek obr. 13. Oddělení objektů a pozadí, využitelné při dalším zpracování. Prahovací úroveň je vhodné nastavit podle histogramu. Pokročilé algoritmy zohledňují také možnost nevhodného nasvícení nebo další vady. Tyto algoritmy rozdělí snímek na několik částí, pro každou část vytvoří histogram a podle jeho průběhu následně nastaví prahovací úroveň.



Obr. 13 Binární snímek

2.2.2 Detekce hran

Detekci hran lze provést mnoha způsoby, na obr. 14 je použita detekce hran pomocí sobel detektoru. Po provedení zmizí celistvé plochy a vytvoří se pouze hranice mezi objekty a pozadí. Bílé pixely označují, kde se hrana nachází, černé pixely označují zbytek. Za hranu může být považována změna odstínu barvy, směrové nasvícení objektů nebo reálné hranice objektu.



Obr. 14 Detekce hran

2.3 Morfologické operace

Morfologické operace slouží k odstranění nebo naopak k doplnění prázdných míst v nalezených objektech. Níže jsou uvedeny pouze dvě základní, a to dilatace a eroze. V praxi je možné tyto funkce navzájem kombinovat. Tím vzniknou operace uzavření nebo otevření. Liší se pouze pořadím, ve kterém se provede dilatace a eroze.

2.3.1 Dilatace

Dilatace je funkce vhodná pro odstranění šumu v obraze, spojí malá prázdná místa a nahradí je celistvou plochou. Výsledný snímek obsahuje více celistvé plochy, obr. 15. Využívá se především pro sjednocení ploch nebo rozšíření objektů v obraze.



Obr. 15 Snímek s aplikovanou dilatací

2.3.2 Eroze

Eroze je opačná funkce, než je dilatace, při použití dojde k odstranění malého okolí objektů a šumu ze snímku obr. 16. Využitelnost této funkce je především při snaze rozčlenit souvislý obraz na více objektů.



Obr. 16 Snímek s aplikovanou erozí

3 Použitý standard ITU-R BT.656

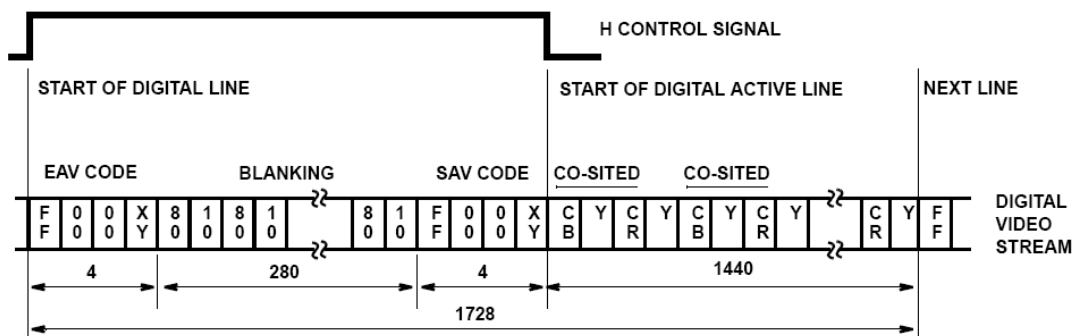
Přenosový protokol BT.656 je často používaný u běžně dostupných převodníků obrazu. Tato standardizace umožňuje efektivně přenášet obrazová data mezi obvody. Standard využívá podvzorkování 4:2:2. Lze přenášet data jak sériově, tak i paralelně. Standard postihuje jak 8 bitový přenos, tak i 10 bitový. Frekvence přenosu dat je 27 MHz.

3.1 Formát dat standardu ITU-R BT.656

Sériový přenos dat je vhodný především u standardních programovatelných obvodů, které mají omezený počet vstupně výstupních pinů, a rychlost zpracování není kritická. Paralelní přenos je využitelný naopak u aplikací, kde přenos a zpracování dat je vhodné co nejvíce zrychlit, a je k dispozici dostatek volných vstupně výstupních pinů pro připojení celé sběrnice.

Standard každý snímek rozděluje na dvě části, na blok složený ze sudých řádků a blok složený z lichých řádků. Mezi každým blokem jsou zatmívací data.

Při paralelním přenosu je k dispozici paralelní sběrnice o šířce 8 nebo 10 bitů, což poskytuje dostatečný datový tok pro přenos obrazu. Dále bude uvažováno pouze o 8 bitové sběrnici. Na obr. 17 je znázorněná přesná posloupnost příchozích dat. Na začátku každého řádku je tzv. SAV blok, který je složen ze čtyř 8 bitových slov. Ten slouží pro detekci začátku řádku.



Obr. 17 Paralelní přenos dat ve formátu BT.656 [1]

Blok je složen z posloupnosti 4 slov, která mají přesně předdefinovaný tvar. První je FF_h , nebo-li 11111111_b , poté dvou nulových slov 00_h , nebo-li 00000000_b a kontrolního slova XY, kterým lze identifikovat, kdy začíná každý řádek příchozího snímku. Složení slova XY je patrné v tab. 6.

Po detekování řádku přicházejí obrazová data, a to v pořadí Cb, Y, Cr, Y, Cb ... pro přenos je efektivnější přenášet obraz ve formátu CbYCr, než v standardně známém formátu RGB, a to z důvodu komprese dat. Při RGB formátu by bylo nutné pro každý pixel přenést 3×8 bitů dat. U formátu CbYCr je využito komprese a přenáší se podvzorkovaný obraz. Lidské oko díky své necitlivosti rozdíl prakticky nezaznamená, ale úspora je pro značná. Jsou k dispozici vždy pro dva pixely 3 informace o rozdílu mezi červenou a modrou barvou a informaci o světlosti obou pixelů. Dohromady 5×8 bitů na dva a půl pixelů, u RGB formátu by pro přenos bylo nutné použít 6×8 bitů na pouhé dva pixely.

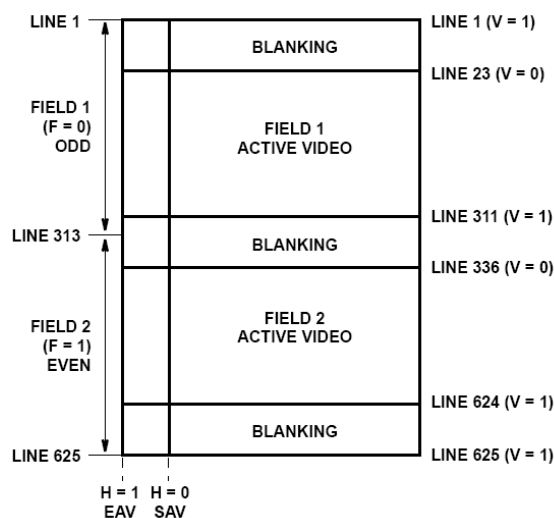
Tab. 6 Popis dat při SAV (XY) a EAV (XY) sekvenci (1)

bit	označení	význam
D₀ LSB	1	trvale nastaven do log. 1
D₁	F	0 – řádek z bloku 0 1 – řádek z bloku 1
D₂	V	1 – zatmívací data
D₃	H	0 – SAV 1 – EAV
D₄	P ₃	$V \oplus H$
D₅	P ₂	$F \oplus H$
D₆	P ₁	$F \oplus V$
D₇ MSB	P ₀	$F \oplus V \oplus H$

Po přenosu daného počtu informací o pixelu následuje EAV blok, který je velmi obdobný jako blok SAV. Rozdíl je patrný z obr. 17. Za tímto blokem následuje sekvence 8 bitových slov s hodnotou 80_h a 10_h, které slouží pro zatmívání. Při dekódování dat přicházejících ve standardu BT.656 nestačí pouze, najde-li se začátek a konec řádku. Aby bylo možné poskládat celý snímek, chybí informace, kde se načtený řádek nachází ve snímku. Z obr. 18 respektive tab. 7 vyplývá, jaké kombinace kontrolního slova XY ukazují na jaký řádek. Například při hledání prvního řádku v celém obrázku, bude vyhledáván řádek, který bude mít ve svém SAV kontrolním slově 1010 1011_b. (poslední tři bity dle tab. 7). Přesný popis kombinací XY je zařazen tematicky v kapitole 5.1.

Tab. 7 Rozložení jednoho snímku [1]

číslo řádku	F	V	H (EAV)	H (SAV)
1-22	0	1	1	0
23-310	0	0	1	0
311-312	0	1	1	0
313-335	1	1	1	0
336-623	1	0	1	0
624-625	1	1	1	0

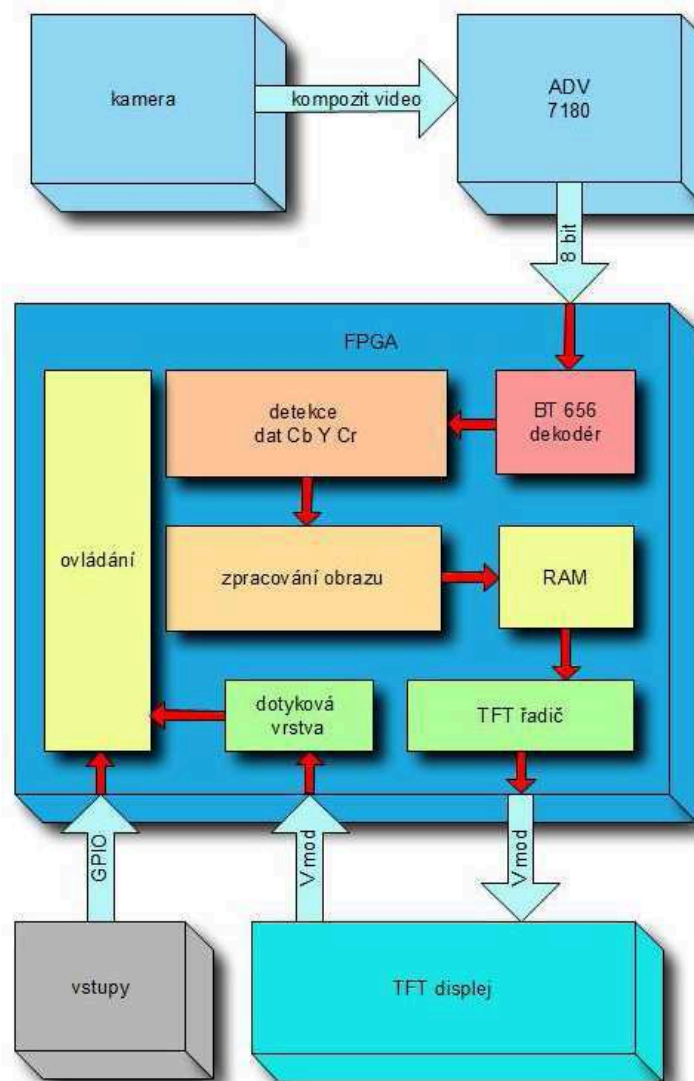


Obr. 18 Rozložení jednoho snímku [1]

4 Koncepce úlohy

Obraz poskytovaný kamerou ve formátu kompozitního videa je přiveden do video převodníku založeného na obvodu ADV7180. Převodník převede analogový signál do kontinuálního digitálního toku dat. Signálů je zpracováván v programovatelném hradlovém poli a vykreslován na dotykovém displeji. Z obr. 19 vyplývá struktura.

Pro úlohu byl zvolen vývojový kit Nexys 3 osazený FPGA Sparatan 6 od výrobce Xilinx. Vývojový kit obsahuje běžné periferie jako VGA, sedmi segmentový displej, LED diody, přepínače a tlačítka. Kromě běžných periférií je kit vybaven konektorem Digilent VHDCI (ve zkratce Vmod), který je využit k připojení dotykového TFT displeje.



Obr. 19 Blokové schéma znázorňující koncept úlohy

V FPGA je implementováno několik bloků, kde jsou postupně vstupní data převáděna a upravována do požadovaného tvaru.

4.1 Kamera Canon MV530

Kamera Canon MV530 obr. 20 je použita jako zdroj video signálu, který bude zpracováván v úloze. Lze ji nahradit jakýmkoli zdrojem analogového obrazového signálu kompatibilního s modulem ADV7180. Kamera poskytuje analogový kompozitní signál. Formát odesílaného signálu je CCIR PAL. Propojení s video převodníkem je realizováno kompozitním videem.



Obr. 20 Kamera MV550 [17]

4.1.1 CCIR PAL signál

CCIR systém byl navržen pro televizní vysílání. Popisuje parametry přenosu obrazového signálu tab. 8 prostřednictvím standardu PAL. Popis standardu PAL 625/50 označuje 625 řádků a 50 bloků. Na zobrazení jednoho snímku jsou potřebné dva bloky. Proto skutečná snímkovací frekvence bude 25 Hz (25 snímků ze sekundu). Počet řádků také není zcela jasný. Standard PAL, posílá 625 řádků, V této hodnotě jsou připočítány i řádky určené pro zatmívání. Řádků s aktivním videem je pouze 576.

Tab. 8 Parametry video signálu kamery MV550

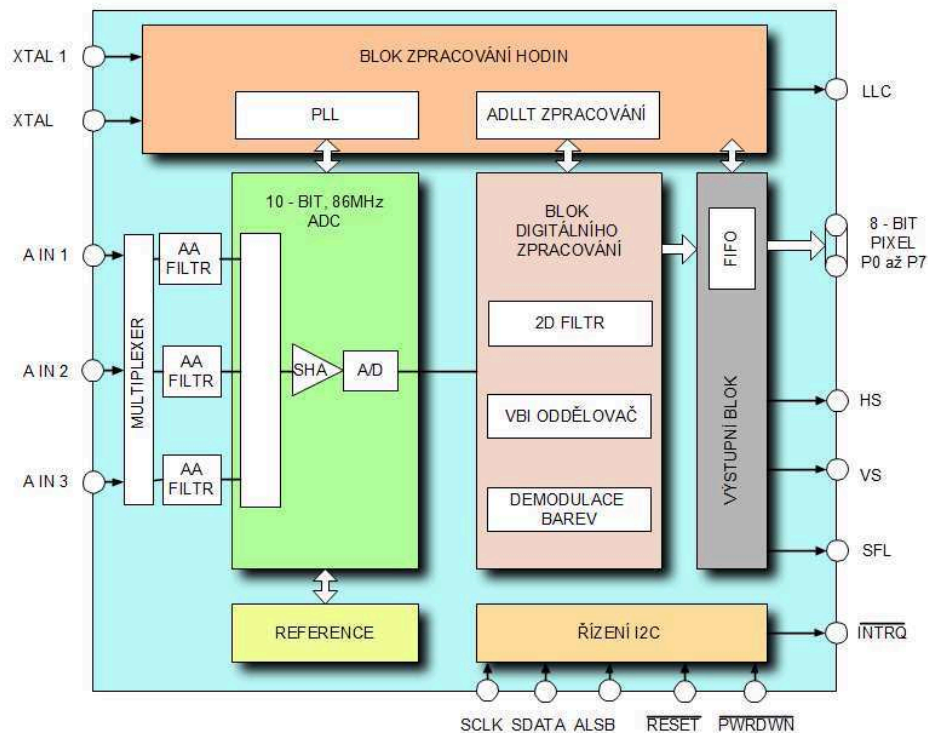
řádky na snímek	576 + zatmívání (celkově 625)
pixely na řádek	704 nebo 720
snímek	25
blok snímku	50

4.1.2 Kompozitní video

Výstup videa je na kameře realizován stereo video kabelem. Ten obsahuje tři signálové vodiče. Levý a pravý kanál zvuku a jeden signálový vodič pro přenos videa. Video standard může přenášet snímek v rozsahu 480 nebo 579 řádků na snímek. Obraz je přenášen analogově. Přenášený obraz je o několik pixelů větší, než je zobrazení na displeji kamery.

4.2 ADV modul

Modul s integrovaným obvodem ADV7180 slouží pro převod analogového video signálu do digitální podoby. Na obr. 21 je znázorněno vnitřní blokové schéma obvodu ADV7180.

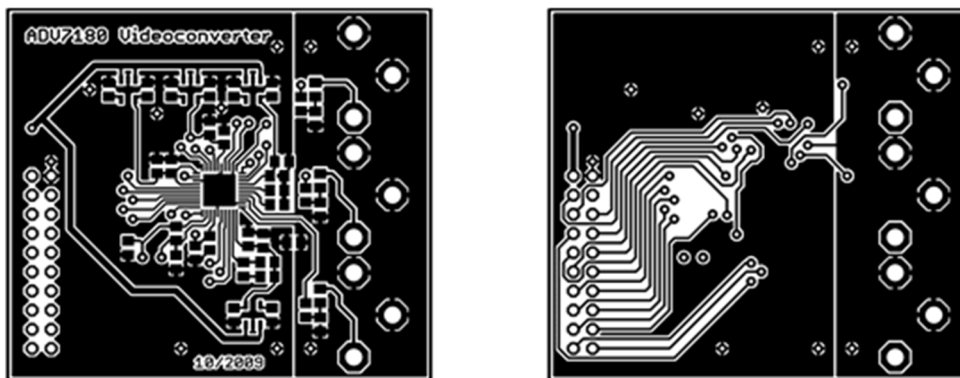


Obr. 21 Blokové schéma IO ADV 7180 [18]

PLL.....fázový závěs
 ADLLT ZPRACOVÁNÍpatentovaný algoritmus pro sledování délky řádku ve video signálu
 AA FILTRantialiasing filtr
 SHAtvarový filtr
 A/Dpřevodník analogového signálu na digitální
 2D FILTRadaptivní filtr oddělující jasovou a barevnou složku obrazu
 VBI oddělovačoddělení obrazových dat od teletextu
 DEMODULACE BAREV ..odděluje jasovou a barevnou složku (PAL nebo NTSC)
 FIFOvyrovnávací paměť

Obvod ADV7180 obsahuje 3 analogové vstupy, které jsou opatřeny antialiasing filtrem. Vlastním fázovým závěsem vytváří signál o frekvenci 86 MHz, kterým vzorkuje vstupní analogový signál. Z digitalizovaného signálu jsou odděleny teletextová data, je oddělena jasová (luma) a barevná (chroma) složka. Poté jsou data dle standardu ITU BT.656 zapsána na výstup. Celý obvod je možné řídit pomocí sběrnice I2C.

Integrovaný obvod je osazen na desce tištěných spojů spolu s referenčním hodinovým zdrojem a napěťovými zdroji obr. 22. Pouzdro obvodu je typu LFCSP se 40 piny. Modul obsahuje tři CINCH konektory pro připojení analogového signálu a dvaceti pinový konektor pro plochý kabel k připojení paralelní sběrnice s daty, I²C komunikaci, napájení a další řídicí signály.

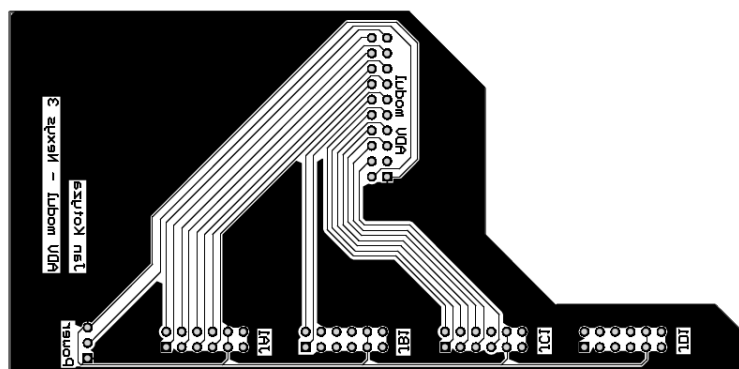


Obr. 22 DPS modulu s ADV 7180 vlevo TOP vpravo BOTTOM vrstva [20]

Napájení modulu je realizováno přes rozhraní USB. Na desce jsou umístěny čtyři stabilizátory. První vytváří napětí 3,3 V sloužící jako referenční napětí pro vstupní a výstupní piny integrovaného obvodu ADV7180. Další dva vytváří napětí 1,8 V pro zpracování vstupního analogového a výstupního digitálního signálu. Poslední stabilizátor slouží pro napájení ostatní logiky.

Referenční hodinový signál pro převodník je vytvořen oscilačním obvodem složeného s krystalu pro frekvenci 28,63636 MHz, dvou kapacit a rezistoru.

Pro připojení modulu byla navržena a vyrobena propojovací deska sloužící jako konstrukční prvek a redukce pro vývojový kit Nexys 3 obr. 23. Snahou bylo zamezit nežádoucím efektům, které by mohly vznikat při připojení vodiči, zlepšit mechanickou pevnost a zvýšit přenositelnost zapojení.



Obr. 23 DPS pro připojení modulu s ADV7180 ke kitu Nexys 3

V tab. 9 je znázorněno propojení modulu s ADV7180 s kitem Nexys 3. Na port JA je přivedena paralelní 8 bitová sběrnice vstupních dat. Zbylé signály jsou přivedeny na porty JB a JC.

Tab. 9 Propojení ADV7180 s Nexys 3

název pinu	ADV 7180	modul s ADV 7180	propojovací DPS	Nexys 3	popis
GND	3, 15, 35, 40	1	GND	GND	GND
VU	-	2	5V	-	5V
VCCO	-	3	-	-	-
PB – ADR0	31	4	ADV4 – JC4	JC4	RESET
PB – DB0	34	5	ADV5 – JC9	JC9	SCLK
PB – ADR1	33	6	ADV6 – JC3	JC3	SDATA
PB – DB1	39	7	ADV7 – JC8	JC8	HS
PB – ADR2	18	8	ADV8 – JC2	JC2	PWRDWN
PB – DV2	2	9	ADV9 – JC7	JC7	SFL
PB-ADR3	37	10	ADV10 – JC1	JC1	VS
PB-DB3	11	11	ADV11 – JB7	JB7	LLC
PB-ADR4	38	12	ADV12 – JB1	JB1	INTRQ
PB-DB4	6	13	ADV13 – JA10	JA10	P6
PB-ADR5	5	14	ADV14 – JA4	JA4	P7
PB-DB5	8	15	ADV15 – JA9	JA9	P4
PB-WE	7	16	ADV16 – JA3	JA3	P5
PB-DB6	10	17	ADV17 – JA8	JA8	P2
PB-OE	9	18	ADV18 – JA2	JA2	P3
PB-DB7	17	19	ADV19 – JA7	JA7	P0
PB-CS	16	20	ADV20 – JA1	JA1	P1

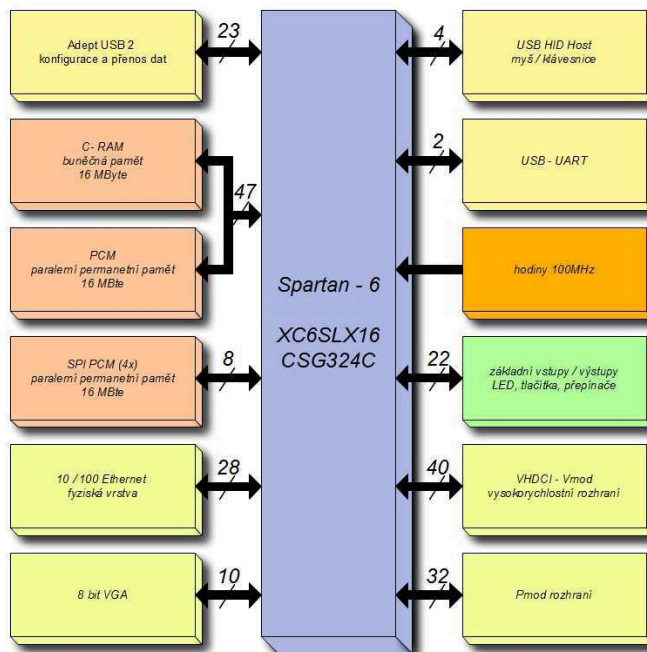
4.3 Vývojový kit Nexys 3

Celá úloha je sestavena a připojena k vývojovému kitu Nexys 3 obr. 24 osazeného výkonným FPGA spartan 6. Vývojový kit obsahuje řadu prostředků pro komunikaci s okolním světem obr. 25.

- pět tlačítek volně dostupných pro uživatele a jedno s funkcí resetu FPGA
- osm přepínačů volně dostupných pro uživatele a jeden pro zapnutí a vypnutí desky
- čtyři osmi segmentové displeje
- čtyři Pmod konektory pro připojení modulů nebo pro volné využití
- VHDCI konektor pro připojení modulů
- USB konektor pro programování a komunikaci přes UART a USB konektor sloužící jako hostitelský, lze ho volně využívat
- Ethernet konektor pro volné využití
- VGA konektor s odporovým DAC převodníkem



Obr. 24 Vývojový kit Nexys 3 [16]



Obr. 25 Blokové schéma vývojového kitu Nexys 3 [16]

4.4 TFT displej

V úloze je použit TFT displej obr. 26 od výrobce Digilent s názvem VmodTFT. Připojení k vývojovému kitu je realizováno přes konektor Digilent VHDCI 2 x 34 pinů, který přenáší informaci v 3.3V LVDS logice. Z toho důvodu je odolnější proti rušení.



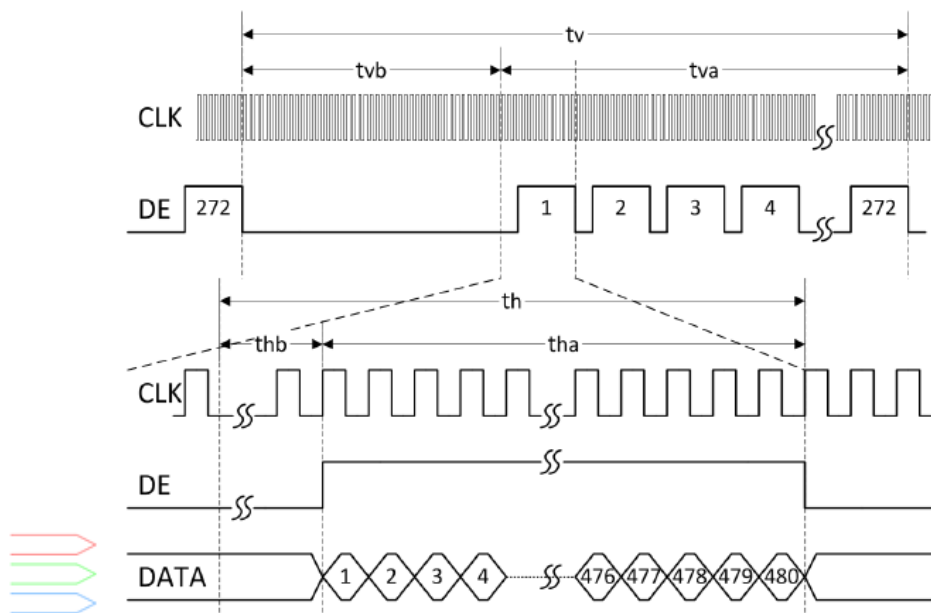
Obr. 26 Vmod TFT displej [15]

Aby bylo možné na displeji cokoli zobrazit, je vhodné mít k dispozici podsvícení a znát samotnou barvu daného pixelu. Podsvícení je řízeno PWM. Barva je s hloubkou 24 bpp. To znamená, že na každý pixel je k dispozici 8 pixelů na červenou, 8 pixelů na zelenou a 8 pixelů na modrou. Displej je vyráběn s odporovou dotykovou vrstvou řízenou integrovaným obvodem AD7873. Řízení toku dat pro vykreslování na displej probíhá signálem *DE* – Data Enabled a hodinovým signálem *CLK*, který může být v rozsahu 7 – 12 MHz. Displej má rozlišení 480 x 272 pixelů tab. 10.

Tab. 10 Typické parametry TFT displeje [15]

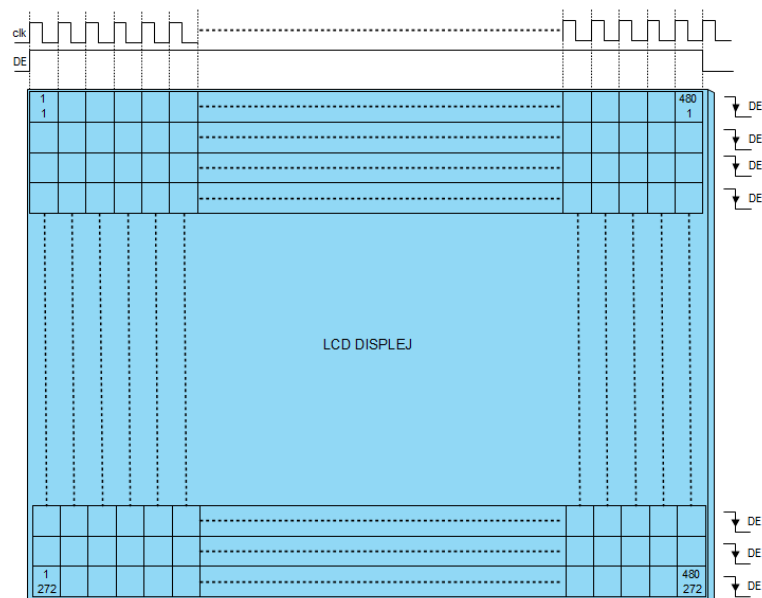
parametr	popis	hodnota	jednotka
fCLK	hodinový signál	9 (7 - 12)	MHz
tVA	perioda vertikálních dat	272	řádky
tVB	perioda vertikálního zatmívání	16	řádky
tHA	perioda horizontálních dat	480	perioda hodin
tHB	perioda horizontálního zatmívání	45	perioda hodin

Na obr. 27 je znázorněn princip kódování výstupních dat pro displej. Po dobu t_{HA} je signál DE v logické 1. Tehdy jsou data s každým pulsem CLK vykreslena na řádek displeje. Za tuto dobu je přenesena 480 krát informace o barvě pixelu. Počet řádků je dán dobou t_{VA} . V době t_{VA} se opakovaně střídají impulsy na signálu DE .



Obr. 27 Časování TFT displej [15]

Každá sestupná hrana signálu DE ukončuje aktuální řádek. Doba, zpožďující odesílání dat následujícího řádku je dána hodnotou t_{HB} , po odeslání 272 řádků následuje konec snímku. Doba mezi jednotlivými snímky určuje t_{VB} . Při dodržení této sekvence lze na displeji zobrazovat až 65 FPS obr. 28.



Obr. 28 Význam signálu pro zobrazení na displeji

4.4.1 Dotyková vrstva

Na horní straně displeje je umístěna dotyková vrstva zaznamenávající stlačení změnou odporu. Obsahuje dvě vodivé průsvitné vrstvy, mezi kterými je tenká mezera. Každá vrstva má přesně definovaný odpor viz tab. 11. Na protilehlých stranách každé vrstvy jsou umístěny elektrody. Při stlačení vrstev dojde mezi deskami k propojení. Bude-li se měřit napětí na elektrodách, lze podle vztahů (4.1) dopočítat souřadnice, kde došlo ke stlačení vrstev.

Tab. 11 Odpor dotykové vrstvy

osa	minimální odpor [Ω]	Maximální odpor [Ω]
X	300	1500
Y	100	900

$$x = \frac{V_x - 96_h}{F6E_h - 96_h} * 480$$

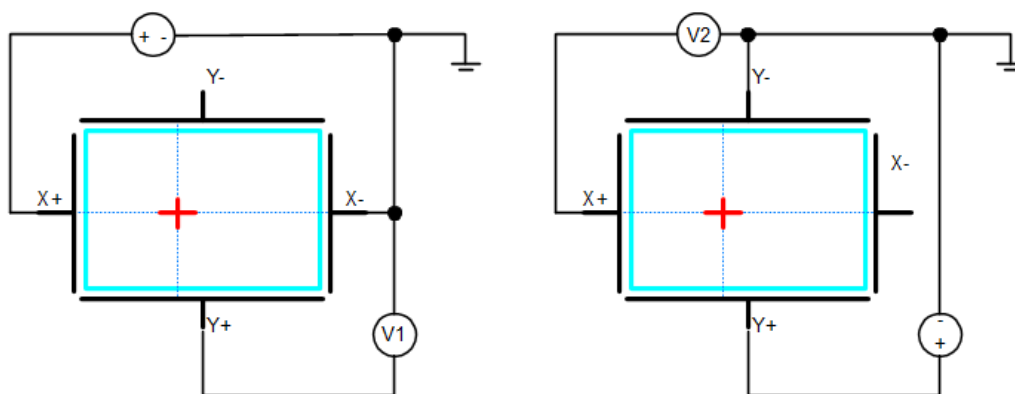
$$y = \frac{V_y - 12C_h}{ED6_h - 12C_h} * 272 \quad (4.1)$$

Proměnné V_x a V_y jsou naměřené hodnoty napětí. Koeficienty objevující se ve vztazích jsou omezující hodnoty odporu pro jednotlivé rohy displeje, jak je patrné v tab. 12.

Tab. 12 12 bitové hodnoty pro kraje dotykové vrstvy [15]

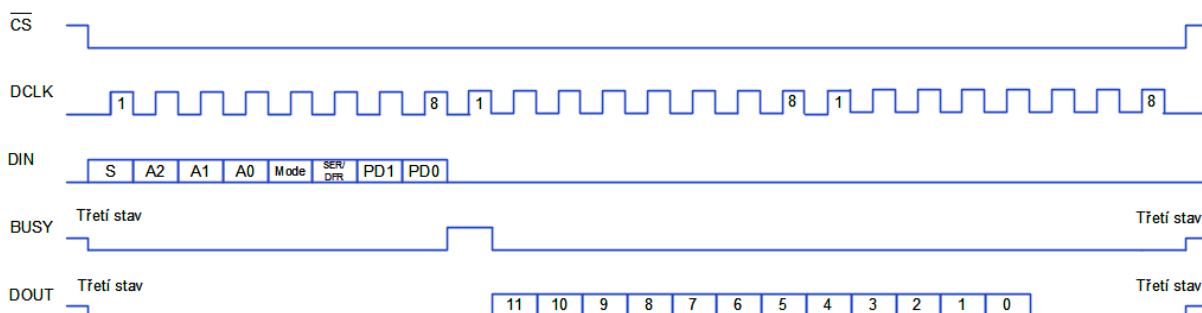
roh	X hodnota	Y hodnota
horní levý	096 _h	12C _h
horní pravý	F6E _h	12C _h
dolní levý	096 _h	ED8 _h
dolní pravý	F6E _h	ED8 _h

Integrovaný obvod ADV7873 slouží pro řízení dotykové vrstvy. Přepíná, které elektrody budou připojeny ke zdroji a na kterých bude probíhat měření. Měření napětí V_x a V_y probíhá dle zapojení na obr. 29. Hodnoty napětí jsou měřeny postupně, první dojde k změření napětí na V1 (V_x) a následně na V2 (V_y). V dalším kroku jsou data sériově odeslána ke zpracování do FPGA.



Obr. 29 Schéma zapojení měřícího obvodu dotykové vrstvy

Sériová komunikace probíhá dle obr. 30. Typický přenos trvá 24 hodinových cyklů. Nejprve je nastaven signál \overline{CS} (Chip select) do logické nuly společně se signálem $BUSY$ a $DOUT$. Poté následuje osm hodinových cyklů, které jsou využity pro přenos stavového registru obr. 31. Hodnoty registru jsou nastavovány signálem DIN podle tab. 13.



Obr. 30 Sériová komunikace s ADV7873 [23]

MSB					LSB		
S	A2	A1	A0	MODE	SER/DFR	PD1	PDO

Obr. 31 Stavový registr obvodu ADV7873 [23]

Tab. 13 Popis stavového registru obvodu ADV7873 [23]

bit	význam	nastavení
S	startovní bit, není-li nastaven, nespustí se přenos	0 – počáteční stav 1 – začátek přenosu
A2 – A0	adresa multiplexeru, nastavující variantu připojení voltmetru dle obr. 29.	001 – měření osy X 101 – měření osy Y
MODE	nastavuje rozlišení výstupních dat	0 – 12 bitové rozlišení 1 – 8 bitové rozlišení
SER/DFR	bit sloužící k nastavení multiplexeru společně s A2 – A0	0 – rozdílový vstup 1 – jednoduchý vstup
PD1 – PD0	slouží k nastavení napájení AD převodníku	11 – trvalé zapnutí ADC

V tab. 13 je popsán význam jednotlivých bitů uvnitř stavového registru pro obvod ADV7873. Nastavení všech kombinací A2-A0 a PD1-PD0 zde není uvedeno, je k dispozici v přílohách nebo v datovém listě [23]. Po ukončení přenosu stavového registru je nastaven signál *BUSY* trvající jeden hodinový cyklus, označující začátek převodu požadované hodnoty. Po nastavení signálu *BUSY* do log nuly následuje sériové zapsání výstupních dat na *DOUT*. Po odeslání dat (v případě na obr. 30 12 bitová varianta) následují tři hodinové cykly, při kterých je *DOUT* nastaven na logickou nulu.

5 Návrh logiky

V návrhu jsou jednotlivé funkční bloky členěny do samostatných modulů, a ty jsou následně deklarovány jako komponenty. Modul „main.vhd“ je tzv. modul nejvyšší úrovně a zastřešuje celý návrh. Kromě VHDL modulů jsou použity také tzv. IP cores, což jsou předem vytvořené a optimalizované funkční celky. Lze je do návrhu vložit pomocí IP generátoru, ve kterém se zvolí požadovaná funkce a pouze se nastaví její parametry. Další částí návrhu jsou soubory obsahující připojení logiky uvnitř FPGA na výstupní piny IO. Soubory mají příponu .ucf.

Návrh celé úlohy byl vytvořen tak, aby byl pokud možno co nejvíce synchronní s řídicím hodinovým signálem. Nicméně v celém zapojení je využito několika vůči sobě asynchronních hodinových domén.

Clk - 100 MHz hardwarově generovaný hodinový signál přivedený na vstup FPGA. Je připojen do DCM a od něj jsou potom odvozeny další hodinové domény

clk_vp - 27 MHz hodinová doména určená pro zpracování příchozích dat z video převodníku

clk_tft - 9 MHz signál pro zapisování zpracovaných dat na TFT displej

clk_tft_180 - 9 MHz.. 180° – signál pro zapisování zpracovaných dat na TFT displej posunutý o 180°

5.1 ITU-BT.656 dekodér

Obrazová data na vstupu FPGA jsou zakódována. Z toho důvodu jsou data dekodována a analyzována. První jsou obrazová data oddělena od zatmívacích dat. Dále probíhá detekce začátku snímku a začátku řádku.

Samotná detekce snímku a řádku probíhá vyhodnocováním specifických příchozích bloků SAV / EAV. Je-li na vstupu detekována sekvence 4 bytů, FF-00-00-XY kde XY nabývá hodnot dle tab. 14, je možné stanovit pořadí aktuálního řádku popřípadě začátek snímku. Z obr. 18 vyplývá, že obrazová data se vyskytují pouze na řádcích 23 – 310 a na řádcích 336 – 623. Tyto řádky jsou zejména důležité.

Tab. 14 SAV a EAV byty

	SAV / EAV	binární tvar								hexadecimální tvar
řádek	XY	1	F	V	H	$V \oplus H$	$F \oplus H$	$F \oplus V$	$F \oplus V \oplus H$	
1 – 22	SAV	1	1	1	0	1	1	0	0	EC
	EAV	1	1	1	1	0	0	0	1	F1
23 – 310	SAV	1	0	0	0	0	0	0	0	80
	EAV	1	0	0	1	1	1	0	1	9D
311 – 312	SAV	1	0	1	0	1	0	1	1	AB
	EAV	1	0	1	1	0	1	1	0	B6
313 – 335	SAV	1	1	1	0	1	1	1	0	EE
	EAV	1	1	1	1	0	0	0	1	F1
336 – 623	SAV	1	1	0	0	0	1	1	1	C7
	EAV	1	1	0	1	1	0	1	0	DA
624 – 625	SAV	1	1	1	0	0	1	0	1	E5
	EAV	1	1	1	1	0	0	0	1	F1

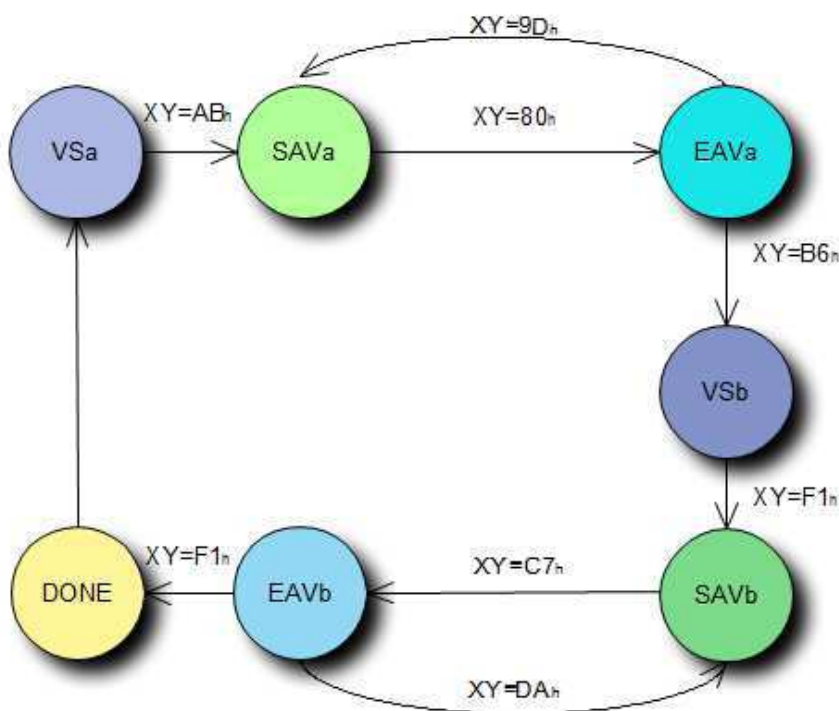
Pro detekci snímků a řádků v příchozích datech byl ve VHDL vytvořen modul, ve kterém se příchozí data ukládají do 5 bytové FIFO vyrovnávací paměti, vytvořené dle následující deklarace ve VHDL (5.1). Dolní čtyři pozice slouží k testování SAV a EAV a z páté jsou vyčítána data. Nastane-li situace, že ve vyrovnávací paměti je kombinace SAV nebo EAV která indikují změnu řádku nebo snímku, modul nastaví data k dalšímu zpracování a nastaví řídicí signál flag o začátku řádku nebo bloku.

```
type FiFo is array (4 downto 0) of std_logic_vector (7 downto 0);
signal VideoBuffer : FiFo; (5.1)
```

FiFo uživatelský datový typ
VideoBuffer signal typu FiFo

type uvozuje, že bude následovat výčtový, uživatelský datový typ
is array označuje, že bude následovat skupina objektů stejného typu

Konstrukce ve VHDL je umístěna v procesu řízeném hodinovým signálem o frekvenci 27 MHz (*clk_cp*) dle video převodníku. Samotná detekce dat probíhá prostřednictvím stavového automatu obr. 32. Realizace ve VHDL je nastíněna na deklaraci (5.2).



Obr. 32 Stavový diagram znázornění funkce detekce dat

Stavy automatu:

- VSa Počáteční stav. Automat čeká na příchod dat s $XY = AB_h$. Tím se detekuje začátek přenosu snímku.
- SAVa Po příchodu $XY = 80_h$ je detekován začátek nového řádku.
- EAVa Data jsou přivedena na výstup bloku. Je vystaven flag o novém řádku. Podle následujícího tvaru XY se přechází buď zpět na SAVa ($XY = 9D_h$) pro čtení dalšího řádku nebo pokračuje na stav VSb ($XY = B6_h$).
- VSb Automat čeká na $XY = F1_h$ pro detekci dalšího bloku dat.
- SAVb Po příchodu $XY = C7_h$ je detekován začátek nového řádku.
- EAVb Data jsou přivedena na výstup bloku. Je vystaven flag o novém řádku a druhém bloku dat. Podle následujícího tvaru XY se přechází buď zpět na SAVb ($XY = DA_h$) pro čtení dalšího řádku nebo pokračuje na stav DONE ($XY = F1_h$).
- DONE Vynuluje se flag o řádku a konci bloku dat a vrací se na stav VSa.

```

type State is (VSa, VSb, SAVa, EAVa, SAVb, EAVb, DONE);
signal VideoState : State := VSa;
:
case VideoState is
  when VSa =>
    :
    when others => null;
end case;

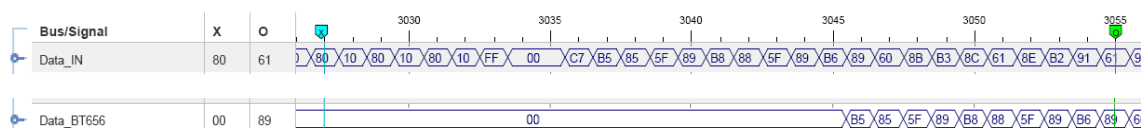
```

(5.2)

State..... uživatelský datový typ
 VideoState..... signál typu FiFo
 VSa ... DONE..... jednotlivé stavy automatu

case-when..... konstrukce stavového automatu, dle aktuálního stavu se vykoná příslušný řádek
 null označuje prázdný příkaz

Na obr. 33 je zobrazen vstupní signál, který přichází z video převodníku na vstup FPGA (sběrnice *Data_IN*) a samotná obrazová data separovaná od ostatních signálů (sběrnice *Data_BT656*). Na sběrnici *Data_IN* jsou zleva znázorněna data s hodnotou 80_h a 10_h , ta slouží jako zatmívání. Za těmito daty následuje blok SAV složený z bytu FF_h dvou bytů 00_h a jednoho bytu ve tvaru XY. Na obr. 33 je XY ve tvaru $C7_h$, podle toho je jasné, že se jedná o řádek v druhém bloku (jeden z řádků 336 – 623 dle tab. 14).

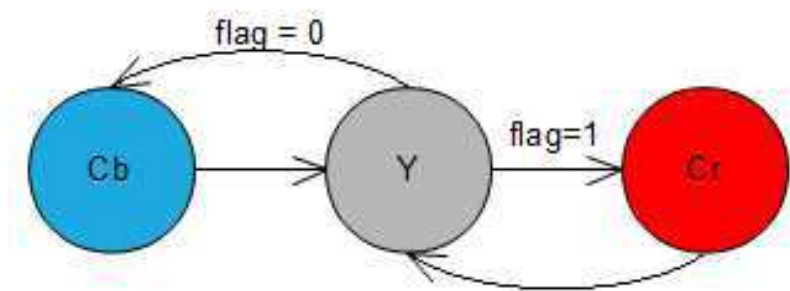


Obr. 33 Záznam vstupních dat z analyzátoru ChipScope pro znázornění funkce modulu BT656

Za bytem XY následují obrazová data ve formátu CbYCr, první byte, který bude dále zpracováván má hodnotu B5_h a reprezentuje signál Cb, následuje 85_h reprezentující složku Y atd...

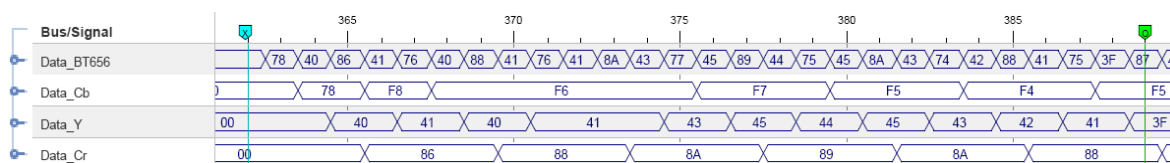
5.2 Detekce dat formátu CbYCr

Třídění dat na jednotlivé složky probíhá jednoduchým stavovým automatem obr. 34. Nový řádek vždy začíná signálem Cb, poté Y, Cr, Y, Cb, tato sekvence se stále opakuje. Rozhodování ve stavu Y je ošetřeno pomocí signálu *c_flag*, který je vždy nastavena do log 1 ve stavu Cb a log 0 ve stavu Cr. Celý automat je řízen pomocí hodinového signálu *clk_vp*.



Obr. 34 Diagram stavového automatu pro detekci složek Cb Y Cr

Na obr. 35 je znázorněna sběrnice *Data_BT656* z předchozího modulu a tři další datové sběrnice, každá obsahující jednu složku z příchozích dat.



Obr. 35 Separace jednotlivých složek

5.3 Zpracování obrazu

V jazyce VHDL není příliš vhodné pracovat s reálnými čísly. Na druhou stranu syntetizátor VHDL vše překládá do dvojkové soustavy. Z toho důvodu se bude provádět úprava koeficientů. Dále bude třeba násobit binární čísla reálnými konstantami. Realizace takového výpočtu bude ve VHDL vypadat dle deklarace (5.3).

```

R : in std_logic_vector (7 downto 0);
:
signal Ras : std_logic_vector (7 downto 0);
:
Ras<=std_logic_vector(to_unsigned((((to_integer(unsigned(R))*a/b),8)));
  
```

(5.3)

R aktuální osmibitová hodnota červené složky jednoho pixelu

Ras pomocný signál, ve kterém bude k dispozici výstupní hodnota
a jmenovatel zlomku, může nabývat jakékoli hodnoty
b..... číselník zlomku, je vhodné, aby byl mocninou čísla dvě
8..... udává bitovou délku výstupního slova ukládanou do *Ras*

Požité konverzní funkce:

std_logic_vector převod přirozeného čísla na osmibitové binární číslo
to_unsigned převod celého čísla na přirozené číslo
to_integer převod osmibitového binárního čísla na celé číslo
unsigned zajišťuje binární hodnotu v *R* jako kladnou (bez dvojkového doplňku)

Výše uvedeným postupem je realizována většina výpočtů pro následující funkce, ve kterých je nutné počítat s reálnými čísly.

5.3.1 Převod do RGB

Převod dat z formátu CbYCr do RGB je realizovaný pomocí logiky umístěné v procesu. Logika je řízena hodinovým signálem *clk_vp*. Je-li při náběžné hraně hodinového signálu impuls o příchodích datech aktivní, provedou se nezbytné matematické operace k výpočtu složek R,G,B. Výpočty proběhnou na základě rovnice (5.4).

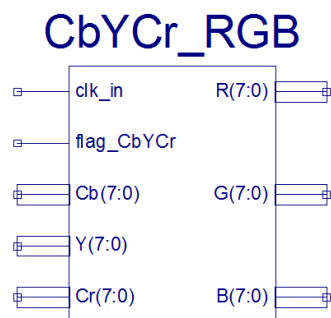
$$\begin{aligned} R' &= Y_{601} + r1 * (Cr - 128) \\ G' &= Y_{601} - g1 * (Cr - 128) - g2 * (Cb - 128) \\ B' &= Y_{601} + b1 * (Cr - 128) \end{aligned} \quad (5.4)$$

Problém ovšem přichází, bude-li nutné na FPGA počítat s desetinnými čísly. Aby tento výpočet proběhl co nejvíce efektivně, koeficienty byly přizpůsobeny programovatelné logice, v tab. 15 jsou koeficienty z rovnice (5.4) upraveny do vhodnější podoby. Desetinné číslo je zapsáno jako podíl celého čísla a *n*-té mocniny čísla dvě. Z toho důvodu dojde k vhodnějšímu využití vnitřní logiky FPGA.

Tab. 15 Koeficienty pro přepočítání složek CbYCr na RGB

	původní hodnoty	upravené hodnoty	zápis celými čísly	chyba
r1	1,371	1,375	11 / 8	-0,004
g1	0,689	0,718	23 / 32	-0,029
g2	0,336	0,343	11 / 32	-0,007
b1	1,732	1,750	14 / 8	-0,018

Modul ve VHDL je reprezentován symbolem na obr. 36. Na vstup bloku jsou přivedeny vstupní hodinový signál, kterými je zpracování řízeno. Signál *flag_CbYCr* slouží pro potvrzení, že se na vstupu nachází platná data. Pod pojmem data se rozumí tři složky barevného signálu, kapitola 1.4.3, ve které jsou popsány. Výstupem bloku jsou přepočítaná data ve formátu RGB. Přepočítání je realizováno dle rovnic (5.4).



Obr. 36 Symbol modulu VHDL pro převod CbYCr do RGB

5.3.2 Převod do odstínů šedé

V některých případech je vhodné obraz předzpracovat. Tím se rozumí obraz např. převést do odstínů šedé. Při složitějších operacích je efektivnější pracovat jen s černobílým obrazem, než s barevným. Při použití černobílého obrazu se manipuluje jen s jednou hodnotou na pixel (například 0 - 255), kdežto u barevného RGB obrazu to jsou hodnoty tři. Jinými slovy snímek v odstínech šedé barvy vzniká snížením barevné hloubky obrazu. V tomto případě dojde ke snížení z 24 bitů na pouhých 8. Při správném přepočtu vznikne věrohodný černobílý obraz. Lze se setkat také se snížením na 4 bity (false contouring)[22], který zvýrazní v obraze jemné přechody barev. Dalším snížením na 1 bit vznikne černobílý (binární) obraz. Jedná se o jeden z více způsobů, jak binární obraz získat. V kapitole 5.3.4 je popsán další způsob.

Pixel do odstínů šedé lze převést dle rovnice (5.5), kde se každá barevná složka RGB vynásobí vlastním koeficientem. Výsledný obraz vznikne vykreslením výsledné hodnoty do všech třech barevných kanálů (RGB). Při realizaci výpočtu na programovatelné logice jsou využity celočíselné ekvivalenty, z důvodu úspory prostředků viz tab. 16.

$$grey = ar * R + ag * G + ab * B \quad (5.5)$$

Tab. 16 Koeficienty pro přepočet složek R G B do odstínů šedé

	původní hodnoty	upravené hodnoty	zápis celými čísly	chyba
ar	0,298	0,281	9 / 32	0,0176
ag	0,587	0,593	19 / 32	-0,006
ab	0,114	0,125	1 / 8	-0,011

Obraz v odstínech šedé je Y složka výstupního signálu z video převodníku. Vstupní data mají jasovou a dvě barvonosné složky. Při výpočtu obrazu v odstínech šedé podle (5.5) je zpětně pouze počítána jasová složka. V návrhu je možnost zobrazit obě varianty, jak originální jasovou složku tak přepočítaný šedo tónový obraz.



Obr. 37 Symbol modulu VHDL pro převod RGB do odstínů šedé

Modul VHDL pro přepočítání do odstínů šedé je na obr. 37. Logika uvnitř je čistě kombinační, z toho důvodu není na vstupu žádný hodinový signál, ke zpracování dat dochází tedy neustále. Vstupní data jsou ve formátu RGB z předchozího modulu a výstupní data jsou pouze přepočítaná data v odstínech šedé. Přepočítání dat je prováděno dle rovnice (5.5).

5.3.3 Převod do odstínů hnědé (sépie)

Pro porovnání je implementována do FPGA také funkce pro převod obrazu do odstínu hnědé barvy, nebo-li sépie. Tato funkce je velmi populární při tvorbě uměleckých fotografií. Převod je realizován opět pomocí kombinační logiky. Matematické výpočty jsou provedeny podle (5.6),

$$\begin{aligned} R_s &= r_a * R + r_b * G + r_c * B \\ G_s &= g_a * R + g_b * G + g_c * B \\ B_s &= b_a * R + b_b * G + b_c * B \end{aligned} \quad (5.6)$$

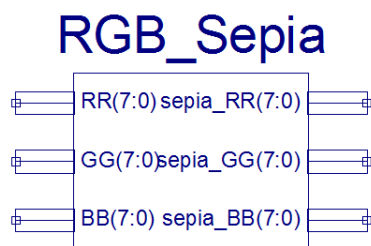
$$sepie = R_s, G_s, B_s \quad (5.7)$$

kde se vypočítá každá barevná složka zvlášť. Výsledný obraz v odstínech hnědé vznikne, jsou-li vykresleny všechny tři upravené barevné složky (5.7). Koeficienty jsou opět převedeny dle tab. 17 pro lepší využití FPGA.

Tab. 17 Koeficienty pro přepočítání složek R G B do odstínů hnědé

	původní hodnoty	upravené hodnoty	zápis celými čísly	chyba
ra	0,393	0,375	3 / 8	0,018
rb	0,769	0,750	3 / 4	0,019
rc	0,189	0,187	3 / 16	0,002
ga	0,349	0,343	11 / 32	0,006
gb	0,686	0,687	11 / 16	-0,001
gc	0,168	0,153	5 / 32	0,015
ba	0,272	0,281	9 / 32	-0,009
bb	0,534	0,531	17 / 32	0,003
bc	0,131	0,125	1 / 8	0,006

Modul pro převod do odstínů hnědé barvy je patrně na obr. 38. Vstupní data jsou opět ve formátu RGB. Pro výsledný obraz je nutné použít všech třech barevných kanálů. Výstupem bloku jsou tedy tři osmi bitové hodnoty.



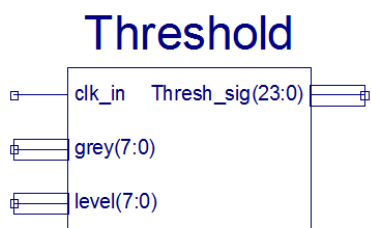
Obr. 38 Symbol modulu VHDL pro převod RGB do odstínů hnědé

5.3.4 Převod do binárního obrazu

Při zpracování obrazu je převod do binárního tvaru běžně používán. Na základě prahovací úrovně je obraz rozdělen na dvě barvy, zpravidla to bývá černá a bílá. Respektive každému pixelu se přidělí hodnota nula nebo jedna, podle aktuální hodnoty v odstínech šedé barvy podle rovnice (5.8) převzaté z [22]. Označení T je použito pro prahovací úroveň, f označuje snímek v odstínech šedé barvy, se kterým se pracuje, g označuje nový černobílý (binární) snímek. Písmeno n udává aktuální porovnávaný pixel. Z toho důvodu mohou z obrazu při správném nastavení zmizet nežádoucí prvky a dostatečně se obraz zjednoduší pro další zpracování.

$$g(n) = \begin{cases} 0, & f(n) \geq T \\ 1, & f(n) \leq T \end{cases} \quad (5.8)$$

Nastavení prahovací úrovně probíhá přes menu na dotykové vrstvě displeje. Stejně tak zapnutí funkce binárního obrazu. K realizaci funkce pro prahování je vhodné použít obraz v odstínech šedé. Funkce je umístěna v procesu a řízena hodinovým signálem. Při každé náběžné hraně se provede porovnání nastavené prahovací úrovně hodnoty aktuálního pixelu v odstínech šedé. Podle výsledku se vykreslí buď jedna nebo druhá barva.



Obr. 39 Symbol modulu VHDL pro převod do binárního obrazu

Na obr. 39 je blok modulu. Příchozí signál *Grey* je porovnáván s nastavenou rozlišovací úrovní. Na jejím základě vznikne více či méně jasný snímek, který bude dále zpracováván.

5.3.5 Hranový detektor

Dalším krokem ve zpracování obrazu může být detekce hran. Ve strojovém vidění se využívá množství způsobů pro detekci hran v obraze. Často využívajících derivací.

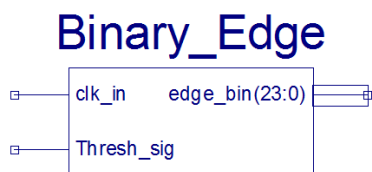
- detekce hran hledáním maxim první derivace (Canny, sobel ...)
- detekce průchodů nulou druhé derivace (D. Marr, E. Hildreth ...)

V této práci bylo využito jednoduché detekce hran z binárního obrazu. V hranovém detektoru je vytvořen dvoubitový posuvný registr, do kterého se ukládá aktuální hodnota jednoho bitu z binárního obrazu. Detekce hran probíhá na základě tab. 18. Je-li v registru přechod z 1 na 0 nebo opačně, jedná se o místo na okraji oblasti a z toho důvodu je zapsána na obrazovku příslušná hodnota.

Tab. 18 Registr pro detekci hrany

stav registru	výsledek
00	pozadí
01	hrana
10	hrana
11	pozadí

Signál *Thresh_sig* se ukládá do posuvného registru s každým hodinovým impulsem *clk_in*. Podle stavu registru se mění výstup signálu *edge_bin* obr. 40.



Obr. 40 Symbol modulu VHDL pro hranový detektor

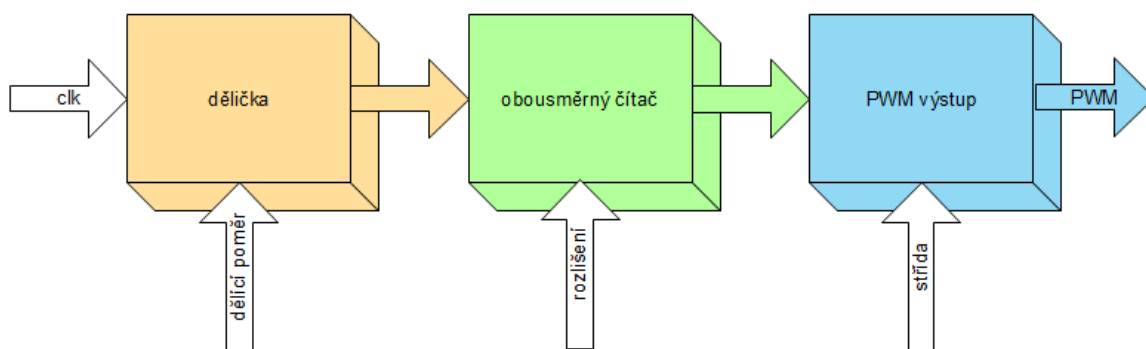
5.4 TFT řadič

Zobrazení na displeji je rozděleno do několika VHDL modulů, nejdůležitější dva jsou modul pro napájení a modul pro časovou základnu.

5.4.1 Napájení TFT displeje

Do této části návrhu patří řízení podsvětlení displeje a startovací sekvence. Podsvětlení je prováděno pomocí generovaného PWM.

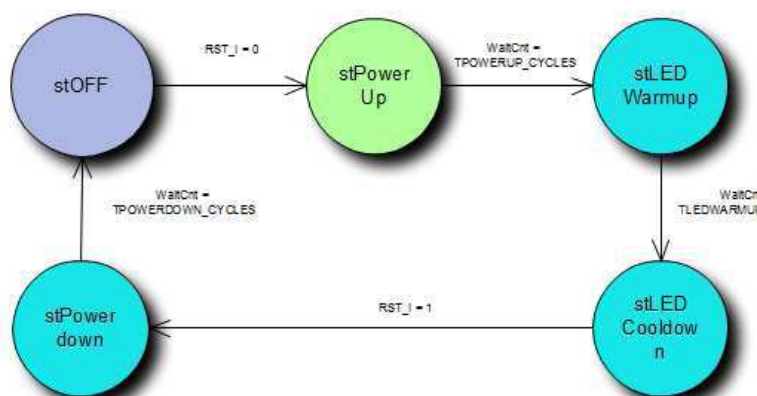
Ve zvláštním VHDL modulu je generováno PWM obr. 41. Ten je pro přehlednost rozdělen do tří na sobě závislých procesů. První proces realizuje děličku hodinového signálu. Výsledkem procesu je vygenerovaný *PWM_enable* signál. Přechází do jedničky, když vnitřní čítač načítá do hodnoty rovné dělicímu poměru. Ten určí délku, respektive frekvenci PWM signálu.



Obr. 41 Blokové uspořádání PWM generátoru

Druhý proces implementuje obousměrný čítač, který přičte nebo odečte jedničku vždy, když je *PWM_enable* signál v jedničce. Tento čítač po dosažení maximální hodnoty rozlišení začne čítat dolů, v nule opět nahoru. Tímto se získá řada hodnot měnící se dle dělicího poměru. Teď již stačí pouze zvolit střidu pro PWM a vždy, když je čítač menší nebo větší než střída, změnit hodnotu PWM výstupu.

Spouštěcí sekvence displeje je realizována stavovým automatem, aby bylo dosaženo sekvenčního chování obr. 42.



Obr. 42 Blokové schéma stavového automatu pro řízení napájení LCD displeje

Stavy automatu:

stOFF počáteční stav ve, ve kterém je displej odpojený
 stPowerUp stav, ve kterém se nastaví bity pro spuštění LCD
 stLEDWarmUp při nastavení stavu stLEDWarmUp se připojí na displej data. Dojde-li k resetu, automat spustí část sekvence pro vypnutí, přechodem do dalšího stavu
 stLEDCooldown stav vytvoří časovou prodlevu pro vypnutí displeje
 stPoerdown..... odpojí displej a přechází do stavu st_OFF

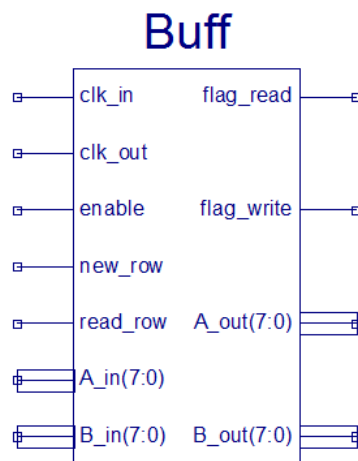
5.4.2 Časová základna

Časová základna je umístěna v jednom procesu. Pro generování vertikálních a horizontálních pulsů jsou využity dvě podmínkové funkce, které jsou do sebe vnořeny. Omezeny jsou hodnotami rozlišení displeje a zatmívacími pulsy, jak je uvedeno v tab. 10. Tím vzniknou impulsy pro synchronizaci a zápis dat na displej.

5.5 Vyrovnávací paměť

Obrazová data jsou načítána s frekvencí 27 MHz a na displej jsou zapisována s frekvencí 9 MHz. Aby nedocházelo ke ztrátě dat nebo jiným chybám, je vytvořena ve VHDL vyrovnávací paměť obr. 43. Ta je sestavena z blokové paměti. Do vyrovnávací paměti se vždy ukládají dva řádky. Řádková vyrovnávací paměť se střídá v zápisu a čtení dat. Délka řádku je zvolena dle vhodného násobku rozlišení obrazu. Rozlišení vstupních dat je až 720 pixelů na řádek. Displej je schopný zobrazit až 480 pixelů na řádek. Z toho důvodu byla použita pouze polovina pixelů ze vstupních dat, a to 360. Zbytek displeje bude využit spolu s dotykovou vrstvou displeje pro ovládání celé úlohy, popřípadě zobrazování informací o zvoleném zobrazení.

Symbol modulu VHDL pro vyrovnávací paměť je znázorněn na obr. 43. Uvnitř modulu jsou již obsaženy vlastní čítače pro ukládání na určitou pozici do paměti. Vstupní hodinový signál slouží pro ovládání rychlosti vyčítání a zápisu. Řídící signály *enable*, *new_row* a *read_row* pro povolení paměti, požadavku na zápis a čtení. Bloková paměť je 16 bitová a vstupní data pouze 8 bitová, proto jsou vytvořeny v návrhu dvě sady vyrovnávacích pamětí. Vstupní data jsou ukládána po dvojicích, tedy složky RG a složky BY, kdy složka Y reprezentuje originální jasovou složku na výstupu video převodníku.



Obr. 43 Symbol modulu VHDL vyrovnávací paměť

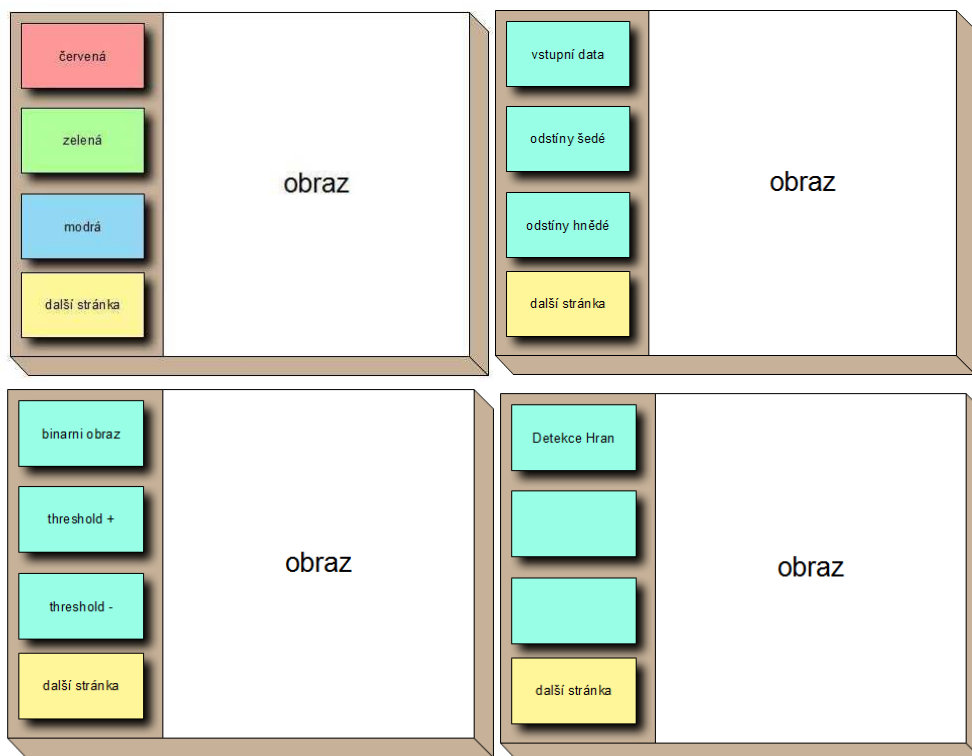
A_in, B_in vstupní data
 flag_CbYCr potvrzení, že jsou na vstupu správná data
 write_row příkaz k zápisu řádku do paměti
 read_row příkaz pro vyčtení řádku z paměti
 A_out, B_out výstupní data
 flag_write kontrola zápisu

 flag_read kontrola čtení
 clk_in hodinový signál pro zápis do paměti
 clk_out hodinový signál pro čtení z paměti

Vyrovnávací paměť je vytvořena z blokové paměti RAM. Paměť je dvouportová. Z toho důvodu je možné do paměti zapisovat a současně z ní vyčítat data. Realizací dvou řádkové paměti se docílilo možnosti pracovat s rozdílnými rychlostmi zápisu na displej.

5.6 Uživatelské rozhraní pro ovládání úlohy

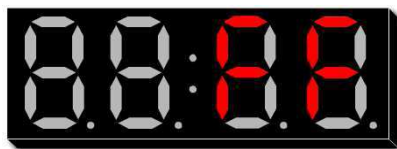
Nastavování zobrazovaných dat probíhá výběráním jednotlivých možností přes dotykovou vrstvu na TFT displeji. Menu je rozčleněno na několik stránek. Na každé jsou čtyři tlačítka. První tři tlačítka slouží ke změně zobrazení a poslední přepne stránku. Po poslední stránce následuje stránka první.



Obr. 44 Ovládací menu

Rozložení obrazovky je patrné z obr. 44. Na první stránce je možnost vykreslit obraz v červené, zelené nebo modré barvě, popřípadě jejich kombinaci. Na druhé straně lze zobrazit příchozí data přímo z video převodníku. Vykresluje se jasová složka ve všech třech kanálech. Další možností je vykreslit obraz v odstínech šedé, vypočítaný z RGB obrazu. Pro porovnání je zde i funkce pro převod obrazu do odstínů hnědé. Na třetí straně je možnost zobrazit binární obraz s možností nastavit prahovací úroveň. Na čtvrté stránce je možnost zapnutí hranové detekce.

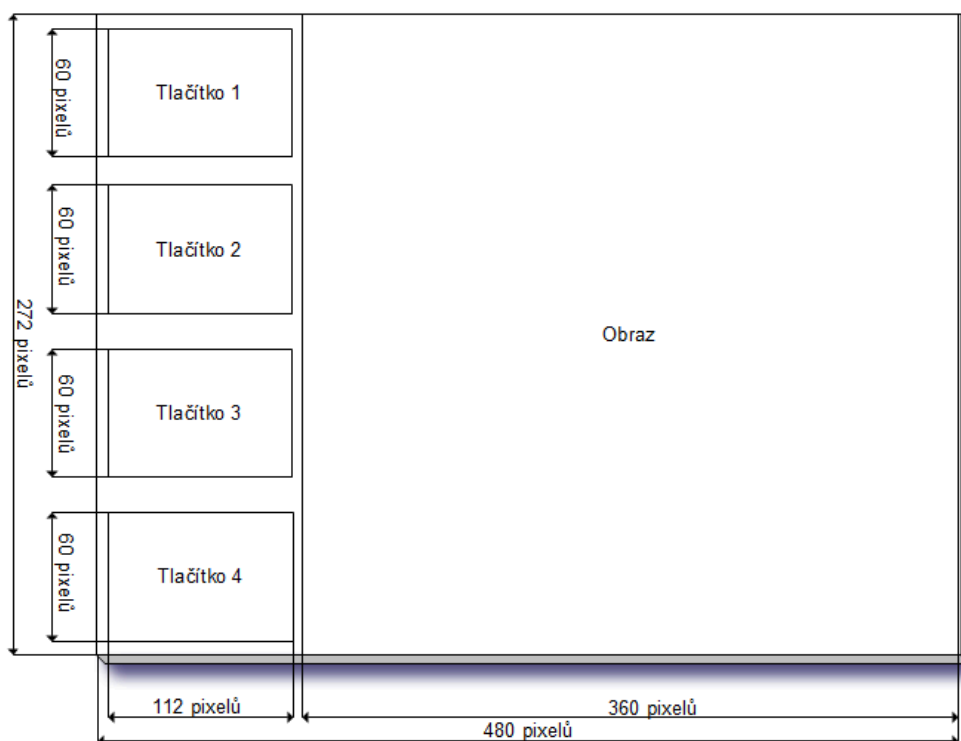
Hodnota prahovací úrovně je přivedena do VHDL modulu, provádějící vypisování hodnoty na sedmi segmentový displej umístěný na vývojovém kitu obr. 45. Algoritmus je umístěn v procesu a z toho důvodu je řízen hodinovým signálem, a to *clk_TFT*. Hodnota je vypisována na dva segmenty ze čtyř. Hodnota je zapsána v hexadecimální soustavě.



Obr. 45 Čtyřmístný sedmissegmentový displej

5.6.1 Realizace tlačítek na dotykové vrstvě displeje

Ovládání systému je možné pouze přes krajní část dotykové vrstvy, na které jsou rozloženy čtyři oblasti citlivé na stlačení. Jejich funkce se mění podle aktuální stránky menu. Jsou umístěny 4 – 116 pixelů od levého okraje obrazovky. Kolem každého tlačítka je oddělovací rámování 4 pixely široké. Výška každé oblasti je 60 pixelů obr. 46.



Obr. 46 Informační rozložení displeje

Po zaznamenání stlačení některé z oblastí se spustí algoritmus pro tlačítko. Algoritmus provádí negaci aktuálního stavu při každém vstupním impulsu, tedy funkce odpovídá spíše představě přepínače. Uvnitř je ošetřeno tak, aby se neprojevily překlmy, respektive opakované rychlé stlačení nebo záchvěvy rezistivní fólie. Ošetření je provedeno zpožděním. Dále algoritmus vzájemným blokováním zajišťuje ochranu proti stlačení více než jedné oblasti v jeden okamžik. Obsahuje také možnost manuálně resetovat hodnotu a vyřadit funkci přepínače.

6 Diagnostika vybraných modulů v FPGA

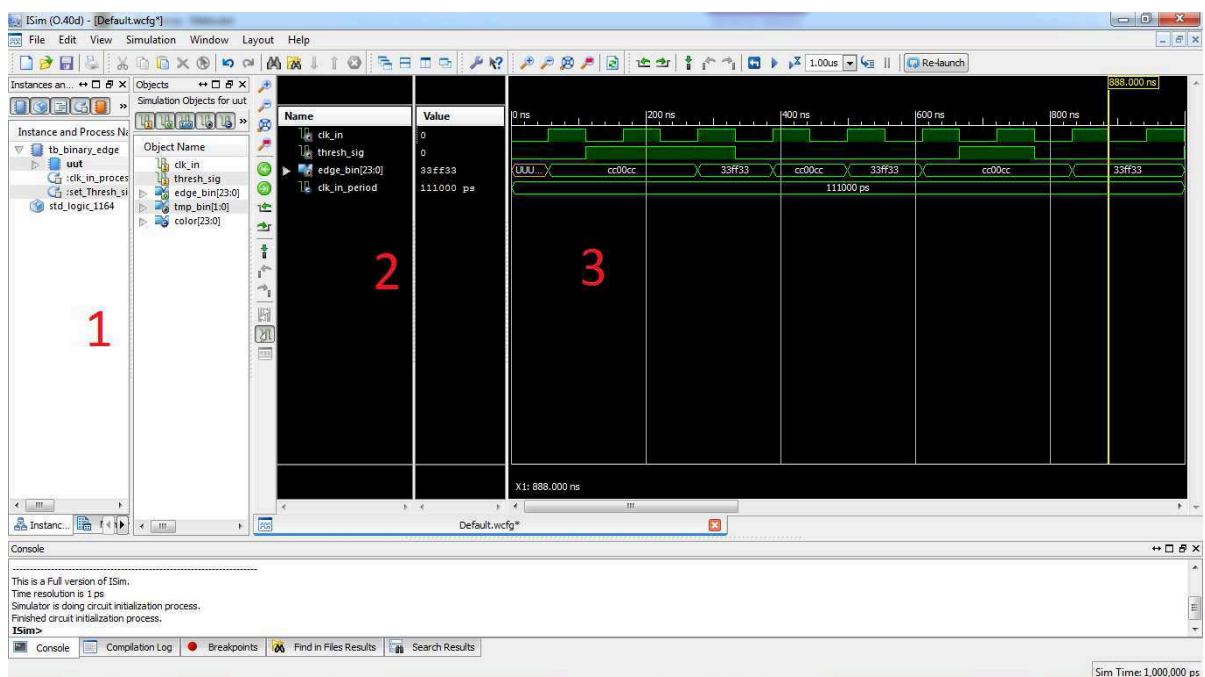
Při tvorbě programu byly vytvořeny základní stavební bloky, které jsou otestovány jak v simulačním prostředí, tak reálně na programovatelné logice. Bloky pro testování byly voleny dle budoucí možné využitelnosti. Otestování ostatních bloků vyšší hierarchické úrovně proběhlo ověřením správné funkce systému.

Vzhledem k rozsahu testování a simulaci každého zvoleného modulu byly vytvořeny dokumenty obsahující informace, popisující každý samostatný blok, včetně výsledků všech simulací a testování. Dokumenty jsou k dispozici v přílohách včetně ukázkových programů obsahujících testovací soubor pro danou simulaci.

Zpoždění jednotlivých bloků je počítáno od času, ve kterém se objeví platná data do času, kdy se platná data projeví na výstupu. Doba zpoždění je přímo závislá na rychlosti hodinového signálu, kterým je obvod případně řízen. Při simulaci jsou zanedbávána zpoždění při průchodu signálu jednotlivými hradly. U různých syntetizačních nástrojů může dojít k rozdílnému návrhu a k různým trasám, přes které signál bude procházet. Z toho důvodu mohou být rozdílné délky zpoždění.

6.1 Simulace

Simulace byly provedeny v simulačním prostředí, viz obr. 47 a tab. 19, dodávaného přímo s instalací ISE. Prostředí slouží převážně pro grafické zobrazení a ověření stavů ve struktuře VHDL modulů. Prostředí ISim využívá pro simulaci testovací VHDL soubor, obsahující deklaraci všech vstupů a výstupů testovaného modulu. Uvnitř testovacího souboru jsou zapsány všechny změny vstupů, které budou postupně měněny.



Obr. 47 Prostředí ISim

Tab. 19 Verze ISim

	jméno	verze	vydání / revize	datum aktualizace
software	ISim	13.1 (nt64)	0.40d	28. únor, 2011

Na obr. 47 je znázorněno spuštěné prostředí. Pro vzorovou ukázkou je v programu otevřen projekt pro binarizaci obrazu. V levé části obrazovky – 1 jsou dva sloupce, obsahující všechny signály a proměnné obsažené v projektu. Ve střední části obrazovky – 2 jsou vypsány signály, které jsou aktuálně zobrazovány. V pravé části – 3 jsou graficky znázorněny hodnoty signálu v závislosti na čase.

V simulačních diagramech jsou oranžově znázorněny některé části signálu. To vyplývá z vlastností jazyka VHDL, který disponuje devíti stavovou logikou tab. 5. Oranžově jsou v simulačním prostředí znázorněny nedefinované hodnoty signálů. Nastávají tehdy, když simulační program nemá počáteční hodnoty. V reálném návrhu je počáteční hodnota buď nastavena při globálním resetu, nebo nezáleží na počáteční hodnotě a při první platně splněné podmínce se hodnota nastaví.

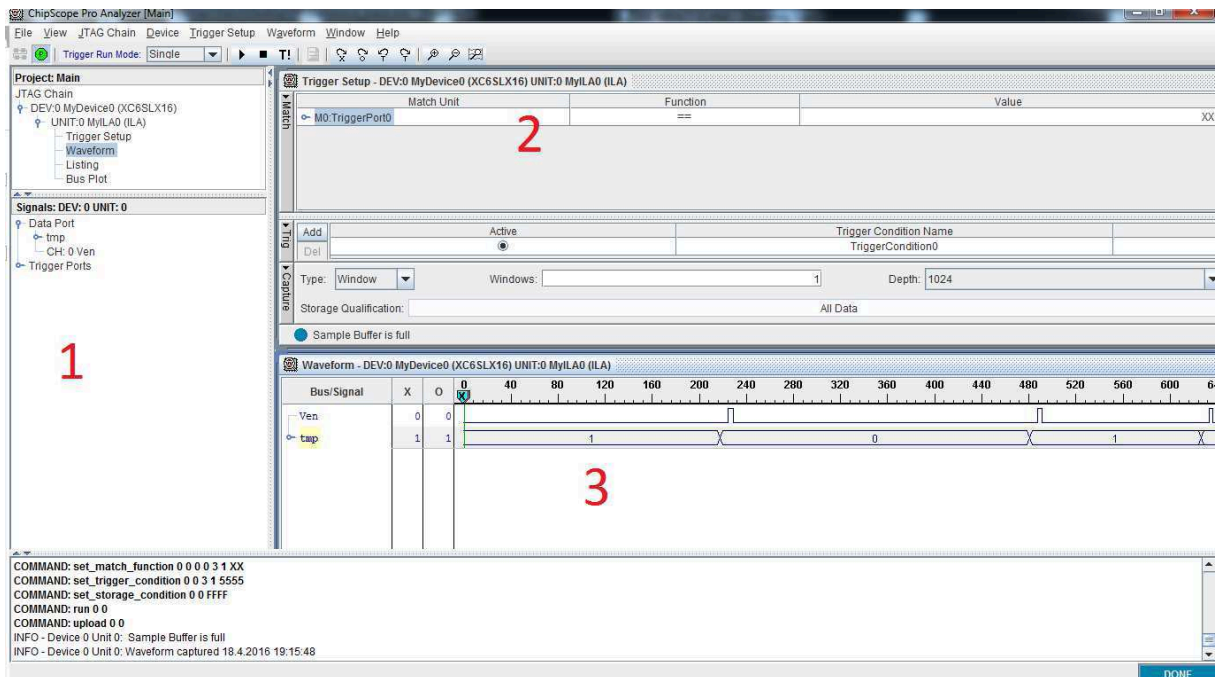
6.2 Testování

Pro testování byl použit logický analyzátor ChipScope tab. 20. Prostředí je dostupné pouze s platnou licencí. Testování jednotlivých bloků probíhalo uvnitř FPGA. Vstupy a výstupy bloků jsou připojeny uvnitř v FPGA k Chipscope. Jedná se o logický analyzátor implementovaný přímo uvnitř navrhované logiky. Z toho důvodu může být připojen a následně testován libovolný signál v navrhované logice. Jsou-li některé vstupy nebo výstupy nepřipojeny, nastaveny na nulu nebo nemění žádným způsobem výstupní signál, jsou signály odstraněny nebo trvale připojeny na požadovanou logickou úroveň.

Tab. 20 Verze ChipScope

	jméno	verze	vydání / revize	datum aktualizace
software	ChiScope pro Analyzer	13.1 (nt64)	0.40d	28. únor, 2011

Na obr. 48 je zobrazeno prostředí pro obsluhu vestavěného logického analyzátoru ChipScope. V levé části obrazovky – 1 jsou zobrazeny signály, které jsou dostupné a připojené k logickému analyzátoru. Vrchní část obrazovky – 2 slouží k nastavení spouštěcích podmínek. Tyto podmínky lze nastavit s velkou variabilitou na libovolný signál. Tato podmínka zajišťuje spuštění analyzátoru pouze v okamžiku, ve kterém probíhá očekávaný děj. Spodní část obrazovky – 3 zobrazuje aktuální hodnotu zvolených signálů závislou na čase.



Obr. 48 Prostředí ChipScope

6.3 Syntéza

Rozvržení bloků na programovatelnou logiku se starají syntetizační nástroje, lišící se u různých výrobců, popřípadě různými architekturami jednotlivých FPGA. Zde použité vývojové prostředí je dodáváno výrobcem xilinx a to ISE 13.1 (nt64) / O.40d.

7 Zhodnocení využití FPGA při zpracování obrazu

Návrh byl realizován se snahou ukázat efektivnost FPGA při práci s velkým množstvím dat. Respektive zpracování videosignálu s minimálním možným zpožděním, tzn., aby se vstupní data zaznamenávaná na kameře projevila v ideálním případě okamžitě na výstupu (displeji). Z toho důvodu by měl být takový návrh využitelný i v praxi, kde by tvořil mezikrok mezi kamerovým čipem a výkonným počítačem, který by ve video signálu z kamery vyhledával například geometrické obrazce, detekoval obličeje, otisky prstů, rozpoznával osoby podle sítnice nebo by vyhodnocoval pohyb na snímaném obraze. Realizovaný návrh je schopný díky vysoké míře paralelismu převzít část kroků při zpracování obrazu s minimální časovou ztrátou. V přílohách jsou umístěny dokumenty popisující zpoždění několika VHDL modulů zpracovávající obrazový signál. Zatímco standartní PC by pro každý krok zpracování potřebovalo desítky hodinových taktů, FPGA stačí jednotky. To by se mohlo zdát jako bezvýznamné ve srovnání s pracovními frekvencemi PC a FPGA, kde PC je schopné pracovat v jednotkách GHz, zatímco FPGA pracuje s horním limitem ve stovkách MHz. Největší výhodu lze spatřit v tom, že PC je schopné provádět výpočty sice rychle, ale sériově, kdežto FPGA může provádět desítky paralelních výpočtů při nižší rychlosti.

Ve výsledném návrhu je největší zpoždění zapříčiněno využitím vyrovnávací paměti, použité pro synchronizaci dat mezi video převodníkem a displejem. Toto zpoždění by v praktickém použití bylo možné částečně redukovat. Zde uvedený návrh výsledná data vykresloval pro názornost na TFT dotykový displej, který využíval pro zápis třetinovou frekvenci, než byla frekvence vstupních dat. Z toho důvodu byla použita vyrovnávací paměť, která data převáděla do pomalejší hodinové domény. V praxi by bylo možné tato data dále posílat přímo do PC nebo jiného nadřazeného systému, popřípadě veškeré obrazové operace a detekci objektů provádět přímo v FPGA, čímž by se ušetřily náklady na další systém. Výstupem z FPGA by mohl být například jednoduchý jednobitový signál informující o přítomnosti objektu (detekce otisků prstů, FPGA srovná načtený otisk s online databází a rozhodne, jestli má osoba oprávnění pro vstup nebo ne).

Samotné obrazové operace realizované na FPGA jako je převod do RGB, odstínů šedé, binárního obrazu nebo hranový detektor jsou pro FPGA triviální a zpoždění dat je u některých téměř nulové, popřípadě několik hodinových taktů, viz přílohy. To souvisí s vysokou mírou paralelismu a možností přistupovat přímo na nejnižší vrstvu, ve které není nutné provádět složité příkazy, například pro přístup do paměti, využití periférií a jejich nastavování nebo obsluhovat přerušení, jak je tomu u standartních PC popřípadě jiných platforem.

Praktické budoucí využití konkrétní zrealizované úlohy je pravděpodobně pouze jako vzdělávací prostředek znázorňující možnosti využití FPGA při zpracování obrazového signálu. Nicméně při realizaci složitějších systémů může být zrealizovaná úloha vhodným zdrojem inspirace a informací. Vytvořené příložené dokumenty mohou pomoci pochopit princip činnosti a implementaci jednotlivých modulů. Popřípadě porozumět principu implementace operací pro zpracování obrazu do obvodů s programovatelnou logikou.

Závěr

V práci byly sepsány jednotlivé kapitoly dle zadání. Teoretická část slouží pro uvedení do problematiky zpracování obrazu jako takového. V teoretické části jsou také nastíněny standardy pro přenos digitální obrazové informace. Navržení demonstrační úlohy proběhlo se snahou navrhnout ucelený přenosový řetězec. Při návrhu struktury úlohy bylo provedeno několik kompromisů. Asi nejvýznamnějším byla volba zdroje digitálního video signálu. Původním záměrem práce bylo využití pouze digitálních a co nejvíce moderních technologií. Při realizaci byla nakonec využita analogová videokamera s video převodníkem.

Návrh logiky probíhal iteračním způsobem. Byl vytvořen návrh funkce, následně byla funkce přepsána do VHDL a testována správná funkčnost. Testování trvalo dlouho z důvodu velké časové náročnosti při syntéze návrhu. Při návrhu nastaly problémy, které komplikovaly průběh návrhu úlohy. Například z původně zamýšlené vyrovnávací paměti pro celý snímek byla realizována pouze dvouřádková paměť. Využití FPGA nenabízí dostatečně rozsáhlou blokovou paměť pro celý snímek. Nabízela se ještě možnost využití externí paměti osazené na vývojové desce, nicméně komunikace s touto pamětí se nepovedla uskutečnit na vyšší frekvenci než 30MHz a při pokusu využít sníženou rychlost komunikace docházelo ke značné ztrátě dat. Dalším problémem se ukázala být synchronizace obrazu na displeji. Z důvodu dvouřádkové vyrovnávací paměti nebylo k dispozici dostatečné množství času pro synchronizaci. Původní záměr byl ukládat celý snímek do paměti a nezávisle na vstupních datech vykreslovat na displej obraz. Nicméně při snížené velikosti paměti tato varianta byla nereálná. Problém se objevil také při dekódování vstupních dat z video převodníku. Data 3 krát až 5 krát za jeden snímek obsahují chybná data v části SAV. Z toho důvodu se nezaznamená nový příchozí řádek a dochází k poškození obrazu.

Výsledkem realizační části práce je systém schopný nasnímat obraz a zpracovat jej uvnitř programovatelné logiky a následně nasnímaný obraz vykreslit na displej. Na základě zpětné vazby od uživatele probíhající přes dotykovou vrstvu displeje je možné změnit způsob vykreslení obrazu na displej.

Dalším možným krokem, jak v práci pokračovat by bylo nalezení vhodné alternativy pro vyrovnávací paměť, popřípadě provádět komplexnější zpracování dat a odesílat je například do PC.

Součástí práce jsou také dokumenty vypracované nad rámec práce, obsahující detailní popis vybraných modulů vhodných pro samostatné využití.

Práce přináší alternativní a velice silný nástroj, jak pracovat s videosignálem. S rostoucím paralelním zpracováním signálů dochází ke značné úspoře času a lze tak snadno realizovat operace, které by na standardních počítačích byly zdlouhavé.

Práce pro mě byla zajímavou možností pochopit a osvojit si způsoby programování FPGA a realizovat složitý systém. Při tvorbě systému jsem se potýkal s častými problémy, souvisejícími s konceptem úlohy nebo také s praktickou realizací navrhované logiky. V neposlední řadě jsem si zde osvojil možnosti pro ladění FPGA, které jsou na rozdíl od standardní elektroniky odlišné. Z toho důvodu považuji práci za velmi přínosnou jak pro mě, tak i pro budoucí čtenáře.

literatura

- [1]. BT.656 Video Interface for ICs: Application Note. *BT.656 Video Interface for ICs: Application Note* [online]. United states: Intersil Corporation, 2002, July 2002 [cit. 2015-11-02]. Dostupné z: <https://www.intersil.com/content/dam/Intersil/documents/an97/an9728.pdf>
- [2]. Ximea: *Products* [online]. [cit. 2015-11-02]. Dostupné z: <http://www.ximea.com/en/products>
- [3]. *SFC Vision systems* [online]. SFCC Vision Systems Product Line, ©2015 [cit. 2015-11-02]. Dostupné z: <http://www.camcables.com>
- [4]. *Wikipedia: The Free encyclopedia* [online]. [cit. 2015-11-03]. Dostupné z: https://en.wikipedia.org/wiki/Main_Page
- [5]. *Digital trends: HDMI VS. DISPLAYPORT VS. DVI VS. VGA* [online]. květen24,2014 [cit. 2015-11-03]. Dostupné z: <http://www.digitaltrends.com/buying-guides/hdmi-vs-dvi-vs-displayport-vs-vga/>
- [6]. *Emsys: FireWire camera* [online]. ©1998-2015 [cit. 2015-11-29]. Dostupné z: <http://www.emsys.de/images/firewire-camera-anon.png/view>
- [7]. *Engadget: GE1910* [online]. 2008, November 30th 2008 At 12:59pm [cit. 2015-11-29]. Dostupné z: <http://www.engadget.com/2008/11/30/prosilicas-ge1910-camera-brings-kodak-hd-sensor-together-with-g/>
- [8]. *Alrad: Imaging* [online]. [cit. 2015-11-29]. Dostupné z: <http://www.alrad.co.uk/imaging/Profile-Sentech.html>
- [9]. *Bioimager: Biocam* [online]. [cit. 2015-11-29]. Dostupné z: <http://www.bioimager.com/id233.html>
- [10]. *Phase1tech: benefits of CoaxPress* [online]. 2015, November 19, 2015 [cit. 2015-11-29]. Dostupné z: <http://blog.phase1tech.com/benefits-coaxpress-standard-high-speed-data-transfer/1774>
- [11]. *Unifircesales: IO industries Cameras* [online]. ©2013 [cit. 2015-11-29]. Dostupné z: <http://www.uniforcesales.com/standard-products/standard-products/cameras/io-cameras/>
- [12]. Taiten: PECL and LVDS OUtputs. *Taiten: PECL and LVDS OUtputs* [online]. [cit. 2015-11-29]. Dostupné z: <http://www.taitien.com.tw/db/pictures/modules/CMS/CMS060207001/AP20100820-PECL-and-LVDS.pdf>
- [13]. *Texas Instruments: Interfacing Between LVPECL, VML, CML, and LVDS Levels* [online]. 2002 [cit. 2015-11-29]. Dostupné z: <http://www.ti.com/lit/an/slla120/slla120.pdf>
- [14]. *Ddpp: Low-Voltage CMOS Logic and Interfacing* [online]. [cit. 2015-11-29]. Dostupné z: http://www.ddpp.com/DDPP3_mkt/c03samp2.pdf

- [15]. Digilent: VmodTFT™ Reference Manual. *Digilent* [online]. 2011, November 11, 2011 [cit. 2015-12-01]. Dostupné z: http://www.digilentinc.com/Data/Products/VMOD-TFT/VmodTFT_rm.pdf
- [16]. Digilent: Nexys 3. *Digilent: Nexys 3* [online]. 2013, April 10, 2013 [cit. 2015-12-02]. Dostupné z: http://www.digilentinc.com/Data/Products/NEXYS3/Nexys3_rm_V2.pdf
- [17]. Canon-europe: Instruction Manual. *Canon-europe: Instruction Manual* [online]. 2002 [cit. 2015-12-02]. Dostupné z: http://files.canon-europe.com/files/soft29721/Manual/MV550i_530i_En.pdf
- [18]. Analog Devices: ADV7180. *Analog Devices: ADV7180* [online]. 2006, 2015 [cit. 2015-12-04]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/ADV7180.pdf>
- [19]. Color Spaces: Chapter 3. *Compression: Everything about the data compression* [online]. Russian, Moscow, 2001, 2014 [cit. 2016-01-05]. Dostupné z: http://compression.ru/download/articles/color_space/ch03.pdf
- [20]. *Návrh a realizace DPS video převodníku s ADV7180 pro FPGA Xilinx Spartan 3: Programování hradlových polí*. Semestrální práce. Vysoká škola báňská – Technická univerzita Ostrava Fakulta elektrotechniky a informatiky. Vedoucí práce Ing. Kašík Vladimír, Ph.D.
- [21]. *Roman Pihan photography page - fotografie a fototechniky: VŠE O SVĚTLE - 5. BAREVNÉ MODELÝ* [online]. ing. Roman Pihan, 2012 [cit. 2016-02-26]. Dostupné z: http://www.fotoroman.cz/techniques3/svetlo05color_model.htm
- [22]. BOVIK, Alan C. [i]Handbook of Image and Video Processing.[/i] 1.vyd. San Diego: Academic Press, c2000. 891 s. ISBN 0-12-119790-5.
- [23]. AD7873: Touch Screen Digitizer Data Sheet (Rev. F). *Http://www.analog.com/en/index.html: Analog Devices* [online]. 2016 [cit. 2016-04-16]. Dostupné z: <http://www.analog.com/media/en/technical-documentation/data-sheets/AD7873.pdf>
- [24]. *Mipi Alliance: About MIPI* [online]. 2016 [cit. 2016-04-20]. Dostupné z: <http://mipi.org/momentum>

Přílohy

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou

- Návod k sestavení úlohy
 - o Navod_k_sestaveni.docx

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ přílohy\
Dokumenty\ Datasheets

- Adresář obsahuje datasheets, manuály a podklady použité v práci.
 - o ADV7873.pdf
 - o ADV7180.pdf
 - o BT656_Video_interface_fot_ICs.pdf
 - o projekt_KL_PHP09.doc

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\
Dokumenty\Knihovny

- Adresář obsahující dokumenty pro vybrané VHDL moduly
 - o 1_CbYCr_RGB.docx
 - o 2_Binary_Edge.docx
 - o 3_RGB_Grey.docx
 - o 4_RGB_Sepie.docx
 - o 5_Threshold.docx

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ Programy\DP

- Adresář obsahující projekt, plnící funkci popsanou v DP

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ Programy\
knihovny\ Binary_Edge

- Adresář obsahující projekt, pro samostatný modul hranového detektoru

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ Programy\
knihovny\ CbYCr_RGB

- Adresář obsahující projekt, pro samostatný modul převodu CbYCr na RGB formát

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ Programy\
knihovny\ RGB_Grey

- Adresář obsahující projekt, pro samostatný modul převodu RGB na do odstínů šedé barvy

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\
knihovny\ RGB_Sepia

- Adresář obsahující projekt, pro samostatný modul převodu RGB na do odstínů hnědé barvy

KOT0099_Zpracovani obrazoveho signalu v obvodech s programovatelnou logikou\ Programy\
knihovny\ Threshold

- Adresář obsahující projekt, pro samostatný modul převodu RGB do černobílého obrazu